

AD-A274 056



AFIT/GE/ENG/93D-28

DTIC
ELECTE
DEC 23 1993
S E D

FAILURE DETECTION, ISOLATION, AND
RECOVERY IN AN INTEGRATED
NAVIGATION SYSTEM

THESIS
William B. Mosle III
Second Lieutenant, USAF

AFIT/GE/ENG/93D-28

93-30997



23997

Approved for public release; distribution unlimited

93 12 22 1 10

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input checked="checked" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



FAILURE DETECTION, ISOLATION, AND RECOVERY IN AN
INTEGRATED NAVIGATION SYSTEM

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

William B. Mosle III, B.S. Electrical Engineering
Second Lieutenant, USAF

December 1993

Approved for public release; distribution unlimited

Preface

This thesis is not the product of any one individual but is a culmination of the efforts of several people. The research topic has been sponsored by the Central Inertial Guidance Test Facility (CIGTF), Holloman AFB, NM which is noted for many contributions to the Air Force's pursuit of well integrated navigation systems. This thesis has been aimed at supporting CIGTF's mission with the study of advanced methods of navigation integration.

This thesis implemented the computer code developed through the efforts of previous AFIT students, Captains Negast, Stacey, Solomon, Snodgrass, and Vasquez. I could not have completed my research without extensive use of the computer software they developed. I would also like to thank John Gustafson of the Avionics Directorate for his help fighting the MSOFE learning curve. The transition of past work to present work could not have been accomplished without the help of Mr. Donald Smith and Captain Neil Hansen, CN Air Force. Their help deciphering and debugging the computer models was indispensable.

I also would like to thank my committee members for aiding me in my research effort. I especially would like to thank LtCol Riggins, my advisor, for his constant support and help throughout the thesis development. I would also like to thank Dr. Maybeck and Captain Delap for their time and effort proof-reading and evaluating my written work.

Thanks goes out to the entire Guidance and Control graduating class of 1993; I cannot imagine working the long hours at AFIT without their peer support. I especially would like to thank 2Lt Odell Reynolds for cajoling me out to the golf course for the occasional sanity check.

Finally I want to thank my fiancé, Jenn, for putting up with all the crap that goes along with getting a Masters. Her presence throughout my last two quarters helped me maintain the proper perspective on my thesis. Now that the thesis is done, I can focus all my attention on our impending matrimony in May.

William B. Mosle III

Table of Contents

	Page
Preface	ii
List of Figures	vii
List of Tables	xiv
Abstract	xvi
 I. Introduction	 1-1
1.1 Background	1-1
1.2 Problem Definition	1-2
1.3 Scope	1-3
1.4 Brief System Description	1-5
1.5 Assumptions	1-7
1.6 Literature Review	1-9
1.6.1 The Concept of Failure Analysis	1-9
1.6.2 The Voting Method	1-11
1.6.3 The Chi-Square Test ($\chi(k)$)	1-11
1.6.4 Generalized Likelihood Ratio Testing (GLR)	1-12
1.6.5 Multiple Model Adaptive Estimation (MMAE)	1-13
1.6.6 A Combination of Techniques	1-15
1.6.7 Literature Review Conclusion	1-17
1.7 Plan of Attack	1-18
1.7.1 Preliminary Research	1-18
1.7.2 System Development and Testing	1-18
1.7.3 Advanced Research into Adaptive Techniques	1-19
1.8 Overview of Thesis	1-19

	Page
II. Kalman Filtering and Failure Detection	2-1
2.1 Overview	2-1
2.2 The Extended Kalman Filter	2-1
2.2.1 The Sampled Data Kalman Filter	2-1
2.2.2 The Discrete-Time Kalman Filter	2-4
2.3 Failure Detection Algorithms	2-5
2.3.1 Chi-Square Equations	2-5
2.3.2 GLR Matching Filter	2-6
2.3.3 The Chi-Square Pattern Recognition(CSPR) Match- ing Filter	2-16
2.3.4 Threshold Selection and Filter Tuning.	2-17
2.4 Chapter Summary	2-17
III. Navigation and Failure Models	3 1
3.1 The MNRS Model Description	3-1
3.2 The NRS Computer Model	3-2
3.3 NRS Model Description	3-4
3.3.1 The Inertial Navigation System(INS) Model.	3-5
3.3.2 The Range/Rauge-Rate System(RRS) Model	3-9
3.3.3 The Global Positioning System(GPS) Model.	3-15
3.4 Failure Models	3-21
3.4.1 Description of the Failures	3-21
3.4.2 Failures in the Truth Measurement Equation	3-23
3.4.3 The GLR Failure Models	3-25
3.4.4 The CSPR Matching Filter Models	3-27
3.4.5 Choosing between the NRS Filters	3-28
3.5 Chapter Summary	3-29

	Page
IV. Results and Analysis	4-1
4.1 The Simulation Specifications	4-1
4.2 Performance of the NRS Filter	4-4
4.3 Performance of the MNRS Filter	4-9
4.3.1 Impact of the Failures on the NRS Navigation Solution	4-10
4.3.2 Results of the MNRS FDIR Algorithm	4-11
4.4 Analysis of Measurement Geometry	4-20
4.5 Performance of the Matching Filter Algorithms	4-23
4.5.1 GLR Matching Filter Results	4-23
4.5.2 CSPR Matching Filter Results	4-26
4.6 Chapter Summary	4-29
V. Conclusions and Recommendations	5-1
5.1 NRS Filter Performance	5-1
5.2 MNRS FDIR Performance	5-2
5.3 Matching Filter Performance	5-3
5.4 Recommendations	5-4
Appendix A. Error State Definitions for the Truth and Filter Models . . .	A-1
Appendix B. Dynamics Matrices and Noise Values	B-1
B.1 Definition of Dynamics Matrices	B-1
B.2 Elements of the Process Noise and Measurement Noise Matrices	B-4
Appendix C. Tuning Plots for an NRS Filter	C-1
Appendix D. Baseline Plots for Residual Monitoring	D-1
Appendix E. Residual Plots for a Transponder Signal Step Failure . . .	E-1
Appendix F. Residual Plots for a Transponder Signal Ramp Failure . . .	F-1

	Page
Appendix G. Residual Plots for a Transponder Signal Noise Failure	G-1
Appendix H. Residual Plots for a Satellite Signal Step Failure	H-1
Appendix I. Residual Plots for a Satellite Signal Ramp Failure	I-1
Appendix J. Residual Plots for a Satellite Signal Noise Failure	J-1
Appendix K. Results of the NRS Filter Chi-Square Tests	K-1
Appendix L. Slope and Intercept Results of the CSPR Matching Filter . .	L-1
Bibliography	B'B-1
Vita	VITA-1

List of Figures

Figure	Page
1.1. How Failures Enter the Navigation System	1-3
1.2. Types of Failures Induced on Measurement Signals	1-4
1.3. NRS System Description	1-5
1.4. MNRS System Description	1-6
1.5. Multiple GLR Testing	1-13
1.6. Multiple Model Adaptive Estimation	1-14
1.7. Combined FDI Algorithm	1-16
3.1. Overall MNRS Description	3-2
3.2. Truth and Filter Model Block Diagram	3-3
3.3. Shape Chi-Square Test as a Result of the Three Failure Types	3-27
4.1. Flight Profile used for the MNRS Simulations	4-2
4.2. Location of the Transponder Transmitters with Respect to the Flight Profile	4-3
4.3. Satellites Orbit Profiles with Respect to the Flight Profile	4-4
4.4. Longitude Filter Error Tuning Plot	4-6
4.5. Example of Scalar Residual Tuning of the NRS Model	4-8
4.6. Relative Tuning of the GPS User Clock Bias and Drift States	4-9
4.7. Comparison of the Effect of Satellite Failures on the Longitude Error State Tracking of NRS Filter 1	4-10
4.8. Baseline Chi-Square Test Results with the Chosen Failure Threshold=15	4-12
4.9. For NRS Filter 1, the Transponder 1 Measurement Residual Response to a Transponder Step Bias=800ft	4-14
4.10. The Effect of a Transponder Ramp Failure on the Residuals of other Satellite and Transponder Measurements	4-15

Figure	Page
4.11. Comparison of Step, Ramp, and Noise Transponder Biases on the Scalar Residual of Transponder Measurement 1 of NRS filter 1	4-16
4.12. Effect of a Satellite Bias as Compared to a Transponder Bias on the Chi-Square Test of NRS Filter 1	4-18
4.13. Comparison of Satellite and Transponder Residuals, Each Affected with a Additive Ramp Bias	4-19
4.14. Comparison of Longitude Error States of NRS Filters with Different Satellite Geometry	4-21
4.15. Comparison of Transponder 1 Scalar Measurement Residuals for NRS Filters 1 and 4 during a Step Bias Simulation Run	4-22
4.16. Comparison of Chi-Square Results for NRS Filters 1 and 4 during a Step Bias Simulation Run	4-22
4.17. Comparison of GLR matching filter results, $\Upsilon(t_i)$, for a Transponder Step, Ramp, and Noise Failure	4-24
4.18. Comparison of GLR Matching Filter Results, $\Upsilon(t_i)$, for a Satellite Step, Ramp, and Noise Failure	4-25
4.19. NRS Filter 1 Chi-Square Line Fit Data for a Satellite Step Bias . . .	4-26
4.20. NRS Filter 1 Chi-Square Results for a Satellite Ramp Failure	4-28
C.1. Latitude and Longitude Error Filter tuning Plots	C-3
C.2. Altitude and North Tilt Error Filter Tuning Plots	C-4
C.3. West and North Tilt Error Filter Tuning Plots	C-5
C.4. North and West Velocity Error Filter Tuning Plots	C-6
C.5. Vertical Velocity and Barometric Altimeter Error Filter Tuning Plots	C-7
C.6. RRS Range and Velocity Bias Filter Tuning Plots	C-8
C.7. GPS User Clock Bias and Drift Filter Tuning Plots	C-9
D.1. NRS1 Satellite Scalar Residual Plots, Baseline	D-3
D.2. NRS1 Transponder Scalar Residual Plots, Baseline	D-4
D.3. NRS2 Satellite Scalar Residual Plots, Baseline	D-5

Figure	Page
D.4. NRS2 Transponder Scalar Residual Plots, Baseline	D-6
D.5. NRS3 Satellite Scalar Residual Plots, Baseline	D-7
D.6. NRS3 Transponder Scalar Residual Plots, Baseline	D-8
D.7. NRS4 Satellite Scalar Residual Plots, Baseline	D-9
D.8. NRS4 Transponder Scalar Residual Plots, Baseline	D-10
D.9. NRS5 Satellite Scalar Residual Plots, Baseline	D-11
D.10. NRS5 Transponder Scalar Residual Plots, Baseline	D-12
D.11. NRS6 Satellite Scalar Residual Plots, Baseline	D-13
D.12. NRS6 Transponder Scalar Residual Plots, Baseline	D-14
D.13. NRS7 Satellite Scalar Residual Plots, Baseline	D-15
D.14. NRS7 Transponder Scalar Residual Plots, Baseline	D-16
D.15. NRS8 Satellite Scalar Residual Plots, Baseline	D-17
D.16. NRS8 Transponder Scalar Residual Plots, Baseline	D-18
D.17. NRS9 Satellite Scalar Residual Plots, Baseline	D-19
D.18. NRS9 Transponder Scalar Residual Plots, Baseline	D-20
D.19. NRS10 Satellite Scalar Residual Plots, Baseline	D-21
D.20. NRS10 Transponder Scalar Residual Plots, Baseline	D-22
E.1. NRS1 Satellite Scalar Residual Plots, Transponder Step Bias	E-2
E.2. NRS1 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-3
E.3. NRS2 Satellite Scalar Residual Plots, Transponder Ramp Bias	E-4
E.4. NRS2 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-5
E.5. NRS3 Satellite Scalar Residual Plots, Transponder Step Bias	E-6
E.6. NRS3 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-7
E.7. NRS4 Satellite Scalar Residual Plots, Transponder Step Bias	E-8
E.8. NRS4 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-9
E.9. NRS5 Satellite Scalar Residual Plots, Transponder Step Bias	E-10
E.10. NRS5 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-11

Figure	Page
E.11. NRS6 Satellite Scalar Residual Plots, Transponder Step Bias	E-12
E.12. NRS6 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-13
E.13. NRS7 Satellite Scalar Residual Plots, Transponder Step Bias	E-14
E.14. NRS7 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-15
E.15. NRS8 Satellite Scalar Residual Plots, Transponder Step Bias	E-16
E.16. NRS8 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-17
E.17. NRS9 Satellite Scalar Residual Plots, Transponder Step Bias	E-18
E.18. NRS9 Transponder Scalar Residual Plots, Transponder Step Bias . . .	E-19
E.19. NRS10 Satellite Scalar Residual Plots, Transponder Step Bias	E-20
E.20. NRS10 Transponder Scalar Residual Plots, Transponder Step Bias . .	E-21
F.1. Transponder 1 Scalar Residual Plots, Transponder Ramp Increase for NRS Filters 1-5	F-2
F.2. Transponder 1 Scalar Residual Plots, Transponder Ramp Increase for NRS Filters 6-9	F-3
G.1. Transponder 1 Scalar Residual Plots, Transponder Noise Increase for NRS Filters 1-5	G-2
G.2. Transponder 1 Scalar Residual Plots, Transponder Noise Increase for NRS Filters 6-9	G-3
H.1. NRS1 Satellite Scalar Residual Plots, Satellite Step Bias	H-2
H.2. NRS1 Transponder Scalar Residual Plots, Satellite Step Bias	H-3
H.3. NRS2 Satellite Scalar Residual Plots, Satellite Step Bias	H-4
H.4. NRS2 Transponder Scalar Residual Plots, Satellite Step Bias	H-5
H.5. NRS3 Satellite Scalar Residual Plots, Satellite Step Bias	H-6
H.6. NRS3 Transponder Scalar Residual Plots, Satellite Step Bias	H-7
H.7. NRS4 Satellite Scalar Residual Plots, Satellite Step Bias	H-8
H.8. NRS4 Transponder Scalar Residual Plots, Satellite Step Bias	H-9
H.9. NRS5 Satellite Scalar Residual Plots, Satellite Step Bias	H-10

Figure	Page
H.10. NRS5 Transponder Scalar Residual Plots, Satellite Step Bias	H-11
H.11. NRS6 Satellite Scalar Residual Plots, Satellite Step Bias	H-12
H.12. NRS6 Transponder Scalar Residual Plots, Satellite Step Bias	H-13
H.13. NRS7 Satellite Scalar Residual Plots, Satellite Step Bias	H-14
H.14. NRS7 Transponder Scalar Residual Plots, Satellite Step Bias	H-15
H.15. NRS8 Satellite Scalar Residual Plots, Satellite Step Bias	H-16
H.16. NRS8 Transponder Scalar Residual Plots, Satellite Step Bias	H-17
H.17. NRS9 Satellite Scalar Residual Plots, Satellite Step Bias	H-18
H.18. NRS9 Transponder Scalar Residual Plots, Satellite Step Bias	H-19
H.19. NRS10 Satellite Scalar Residual Plots, Satellite Step Bias	H-20
H.20. NRS10 Transponder Scalar Residual Plots, Satellite Step Bias	H-21
I.1. Satellite 3 Scalar Residual Plots, Satellite Ramp Bias for NRS Filters 1,2,4,5,6	I-2
I.2. Satellite 3 Scalar Residual Plots, Satellite Ramp Bias for NRS Filters 7-10	I-3
J.1. Satellite 3 Scalar Residual Plots, Satellite Noise Increase for NRS Filters 1,2,4,5 and 6	J-2
J.2. Satellite 3 Scalar Residual Plots, Satellite Noise Increase for NRS Filters 7-10	J-3
K.1. Baseline Run, Chi-Square Test Results for Filters 1-6, N=3	K-3
K.2. Baseline Run, Chi-Square Test Results for Filters 7-10, N=3	K-4
K.3. Threshold Test Results for the Chi-Square Tests on each Filter, Baseline Run	K-4
K.4. Transponder 1 <i>Step Bias</i> =800, Chi-Square Test Results for Filters 1-6, N=3	K-5
K.5. Transponder 1 <i>Step Bias</i> =800, Chi-Square Test Results for Filters 7-10, N=3	K-6
K.6. Threshold Test Results for the Chi-Square Tests on each Filter, <i>Step Bias</i> on Transponder 1	K-6

Figure	Page
K.7. Transponder 1 <i>Ramp Offset</i> =1t, Chi-Square Test Results for Filters 1-6, N=3	K-7
K.8. Transponder 1 <i>Ramp Offset</i> =1t, Chi-Square Test Results for Filters 7-10, N=3	K-8
K.9. Threshold Test Results for the Chi-Square Tests on each Filter, Ramp Offset on Transponder 1	K-8
K.10. Transponder 1 <i>Noise Increase</i> =150, Chi-Square Test Results for Filters 1-6, N=3	K-9
K.11. Transponder 1 <i>Noise Increase</i> =150, Chi-Square Test Results for Filters 7-10, N=3	K-10
K.12. Threshold Test Results for the Chi-Square Tests on each Filter, Noise Bias on Transponder 1	K-10
K.13. Satellite 3 <i>Step Bias</i> =3500, Chi-Square Test Results for Filters 1-6, N=3	K-11
K.14. Satellite 3 <i>Step Bias</i> =3500, Chi-Square Test Results for Filters 7-10, N=2	K-12
K.15. Threshold Test Results for the Chi-Square Tests on each Filter, Step Bias Run on Satellite 3	K-12
K.16. Satellite 3 <i>Ramp Offset</i> =4t, Chi-Square Test Results for Filters 1-6, N=3	K-13
K.17. Satellite 3 <i>Ramp Offset</i> =4t, Chi-Square Test Results for Filters 7-10, N=3	K-14
K.18. Threshold Test Results for the Chi-Square Tests on each Filter, Ramp Offset on Satellite 3	K-14
K.19. Satellite 3 <i>Noise Increase</i> =300, Chi-Square Test Results for Filters 1-6, N=3	K-15
K.20. Satellite 3 <i>Noise Increase</i> =300, Chi-Square Test Results for Filters 7-10, N=3	K-16
K.21. Threshold Test Results for the Chi-Square Tests on each Filter, Noise Bias on Satellite 3	K-16
L.1. NRS Filter 1 Chi-Square Line Fit Data for a Transponder Step Bias	L-2
L.2. NRS Filter 1 Chi-Square Line Fit Data for a Transponder Offset Bias	L-2
L.3. NRS Filter 1 Chi-Square Line Fit Data for a Transponder Noise Increase	L-3

Figure	Page
L.4. NRS Filter 1 Chi-Square Line Fit Data for a Satellite Step Bias . . .	L-3
L.5. NRS Filter 1 Chi-Square Line Fit Data for a Satellite Ramp Offset . .	L-4
L.6. NRS Filter 1 Chi-Square Line Fit Data for a Satellite Noise Increase .	L-4

List of Tables

Table	Page
1.1. MNRS Filter Measurement Structure	1-15
3.1. References for the Sub-Matrices of the NRS Truth and Filter Models .	3-5
3.2. Definition of Failure Variables for each Run	3-24
3.3. Slope and Intercept Values for the Three Types of Failures	3-28
4.1. Location of the Transponders Transmitters in the MNRS Simulation .	4-2
4.2. Location of the GPS Satellites during the MNRS Simulation	4-3
4.3. Temporal Averages of True Filter Errors (1σ)	4-7
4.4. Final Slope and Intercept Values for the Six Failure Runs	4-27
A.1. 39-State INS System Model: First 24 States	A-2
A.2. 39-State INS System Model: Second 19 States	A-3
A.3. 26-State RRS System Model	A-4
A.4. 30-State GPS System Model	A-5
A.5. 15-State Reduced-Order Filter Model	A-6
B.1. Elements of the Dynamics Submatrix F_{Filter}	B-2
B.2. Elements of the Dynamics Submatrix $F_{INS_{11}}$	B-3
B.3. Elements of the Dynamics Submatrix $F_{INS_{12}}$	B-4
B.4. Elements of the Dynamics Submatrix F_{RRS}	B-4
B.5. Elements of the Dynamics Submatrix F_{GPS}	B-4
B.6. Elements of Truth Model Process Noise Submatrix for the INS Truth Model	B-5
B.7. Elements of Truth Model Process Noise for RRS and GPS States . . .	B-5
B.8. Filter Process Noise Q Values	B-5
B.9. Truth and Filter Measurement Noises R Values	B-5

Table	Page
C.1. Legend for Filter Tuning Plots	C-2
D.1. Legend for Filter Tuning Plots	D-2
E.1. Legend for Filter Tuning Plots	E-1
F.1. Legend for Filter Tuning Plots	F-1
G.1. Legend for Filter Tuning Plots	G-1
H.1. Legend for Filter Tuning Plots	H-1
I.1. Legend for Filter Tuning Plots	I-1
J.1. Legend for Filter Tuning Plots	J-1

Abstract

An Inertial Navigation System(INS), the Global Positioning System, and a ground based transponder system(RRS) can all be used to provide the user with a navigation solution. Yet by integrating these three navigation systems with an extended Kalman filter(EKF), a navigation solution is attained that benefits from the information of all three subsystems. This research develops a multiple model EKF failure detection, isolation, and recovery(FDIR) algorithm using a Chi-Square failure test to provide robust navigation solution to measurement failures. The algorithm specifically counters failures in the GPS and RRS range measurements. Analysis is conducted using a Kalman filter development package known as the Multimode Simulation for Optimal Filter Evaluation (MSOFE). Both a large order *truth* model for the navigation system (in which a full 24 satellite constellations is modeled) and a reduced-order Kalman filter are developed. Results suggest that the multiple model algorithm can correct for all single measurement failures.

FAILURE DETECTION, ISOLATION, AND RECOVERY IN AN INTEGRATED NAVIGATION SYSTEM

I. Introduction

Currently within the Air Force arsenal, multiple navigational tools exist to help the pilot navigate his plane. The three navigation tools that will be the focus of this research are the Inertial Navigation System(INS), the Global Positioning System(GPS), and the ground based Range/Range-Rate System(RRS). Each of these tools generates a specific form of information about the position and velocity of the aircraft. The purpose of an integrated navigation system is to collect information from the various navigation tools, and then produce an accurate and robust navigation solution. The navigation solution should consist of the most accurate position, velocity, and orientation estimates for the aircraft. Yet it is desirable for the system to be both accurate and robust. To be robust, the integrated navigation solution must continue to operate as accurately as possible when failures in the system occur. This thesis will focus on a failure detection, isolation, and recovery technique that will make an integrated navigation system robust to GPS and RRS measurement failures.

1.1 Background

The topic pursued in this thesis emanates from ongoing AFIT research. The overall goal of this research initiative is to develop next generation navigation systems for modern aircraft and missile systems. This research has been sponsored by the Central Inertial Guidance Test Facility (CIGTF), 6585th Test Group, Air Force Materiel Command (AFMC), Holloman AFB, NM. CIGTF has sponsored this research in support of their navigation test range.

The past research at AFIT began with the generation of computer models for the INS, GPS, and RRS navigation systems, as well as the development of an integration scheme to blend the information from these three sources into a single navigation solution.

AFIT has chosen to blend the information using an extended Kalman filter (EKF). The overall navigation system developed by this early research became known as the Navigation Reference System (NRS) [14]. Later AFIT research pursued failure detection algorithms to use in conjunction with the NRS, to detect and isolate GPS measurement signal failures [19]. The success of this early failure detection work has led AFIT to pursue a failure detection, isolation, and recovery (FDIR) algorithm that will allow for accurate and robust system performance.

1.2 Problem Definition

The research to be conducted this year will expand on past AFIT research to implement a complete FDIR algorithm for the NRS model. Starting with the original NRS, a new navigation system will be developed that not only can detect and isolate failures, but will also reconfigure to operate accurately during the failures. The failures are errors in the measurement signals received through the GPS and RRS subsystems. The new system that is developed in this thesis will be referred to as the Multiple Navigation Reference System (MNRS).

This thesis will augment past research in three specific areas. First the research will look at failures in both GPS and RRS measurement signals as opposed to past AFIT research which considered only GPS failures [19]. The second addition to this research initiative is the concept of robust navigation. Past research has looked exclusively at failure detection, and isolation (FDI) techniques. This research is adding recoverability, to create a complete FDIR algorithm. This thesis develops one method of modifying the NRS to make it robust to failures. The development of the MNRS moves AFIT one step closer to developing a failure detection, isolation, and reconfiguration algorithm for all possible navigation failures. Finally, this thesis explores various matching filter techniques for identifying the exact type of failure that is affecting the system. Two identification techniques will be evaluated for future applicability to a combined FDIR algorithm.

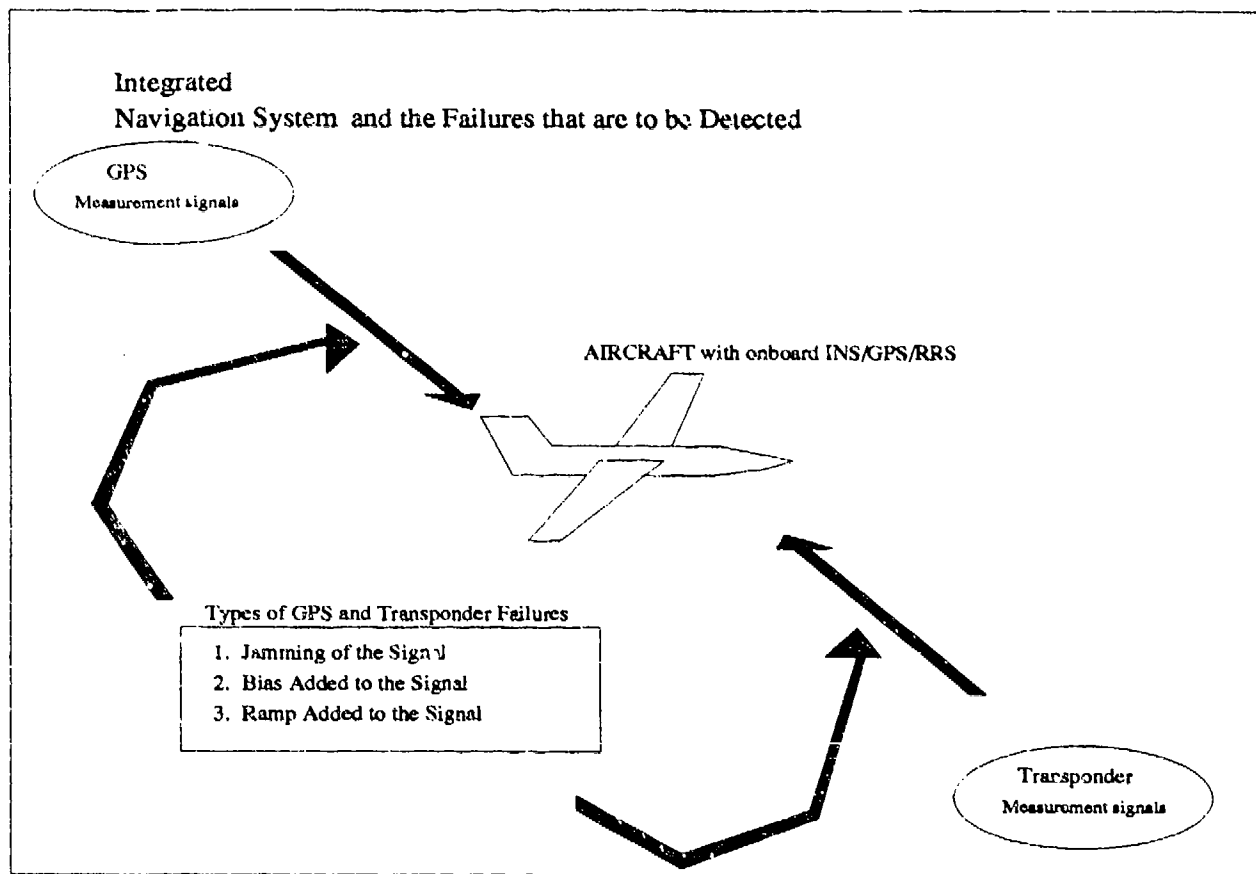


Figure 1.1 How Failures Enter the Navigation System

1.3 Scope

The scope of this thesis is defined by the exact failures that are to be induced within the system. The failures to be considered enter the system through outside measurement sources. The two prevalent navigation measurement sources in the system are the GPS and the RRS measurement signals (see Figure 1.1). Errors in these signals can be caused by natural atmospheric affects, failures in the respective signal transmitters, or hostile enemy signal degradation. Yet the actual source of the failure is not truly important to the FDIR algorithm. Only the effect of the error source on the measurement signal is important for the purpose of FDIR.

This thesis assumes that all the failures in these signals can be modelled as increased noise, a step bias, or a ramp offset added to the original measurement signal. Figure 1.2 shows how each of these failures would affect the uncorrupted GPS or RRS measurement

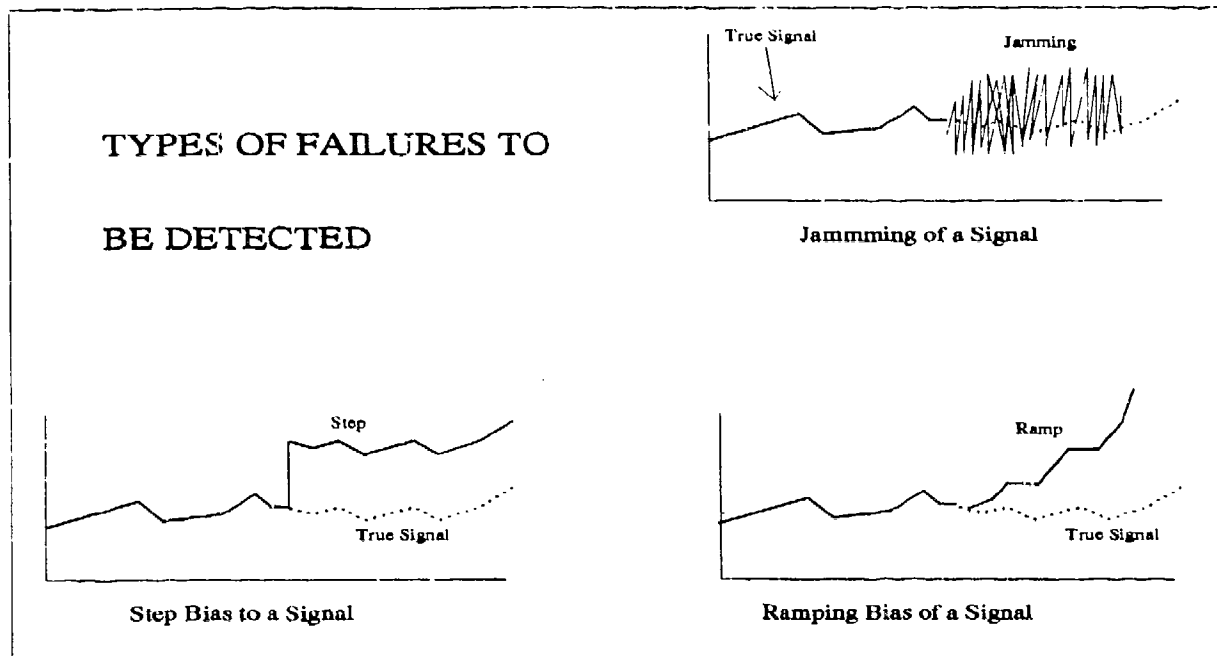


Figure 1.2 Types of Failures Induced on Measurement Signals

signal. The errors that affect the measurement signals are assumed to be additive in nature for this thesis. The use of step and ramp additive failure offsets are standard methods to simulate "spoofing" of a system.

The MNRS is limited to reconfiguring itself for single failures. When two failure occur, the MNRS can detect the existence of the second failure, but cannot recover an unbiased navigation solution. The algorithm also has no ability to isolate the failed measurements when more than one failure occurs. The single failure limitation is mainly due to limited computational modelling capacity. To develop an MNRS system robust to multiple failures would require more parallel computing than is available for this thesis. Therefore, the goal for development of the MNRS system is robustness to single step, ramp, or noise biases induced in either the GPS or RRS measurement signal.

The identification of failure type will focus on distinguishing between step, ramp, and noise offset failures. Two different matching filter techniques are compared to determine which more accurately identifies the type of failure once it has been detected.

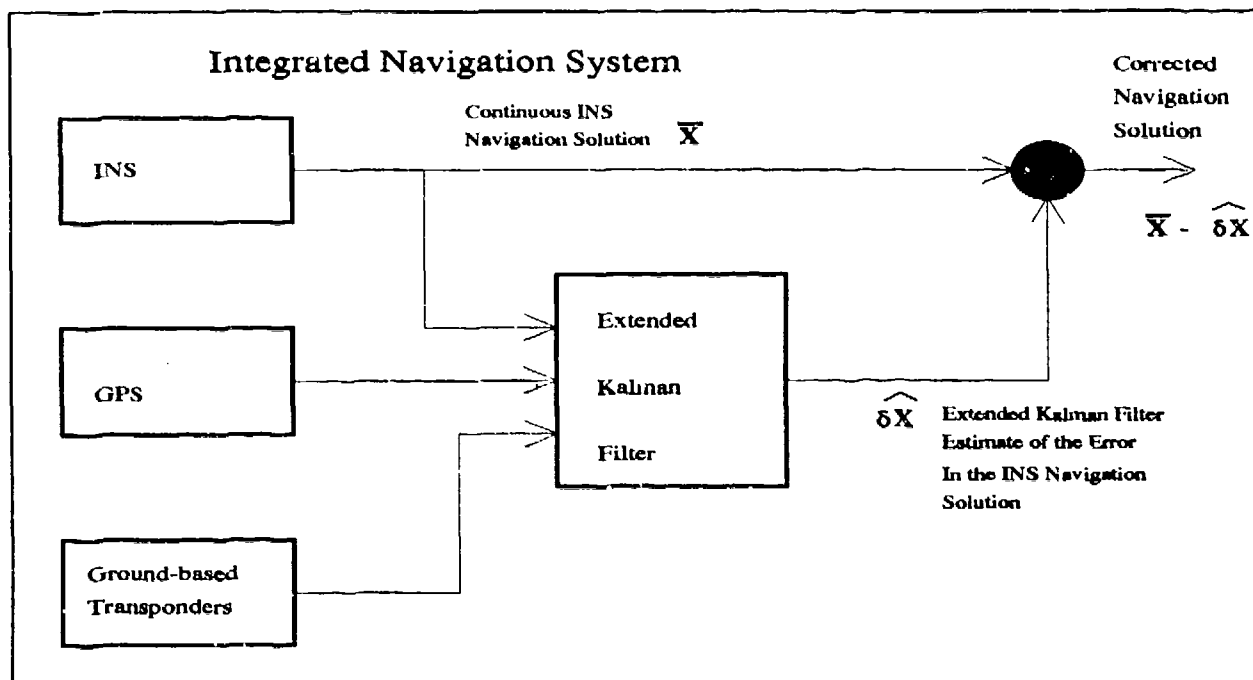


Figure 1.3 NRS System Description

1.4 Brief System Description

The proposed MNRS is based on the work accomplished in past AFIT theses in the development of the NRS. The system description will begin with a description of the original NRS, after which the MNRS will be presented.

The original NRS consists of a single EKF that gathers together information from the GPS, RRS, and INS to produce a single navigation solution. The system is modelled on a Sun SPARC workstation that simulates an aircraft flying a highly dynamic flight profile. The NRS navigation solution is the combination of the continuous INS navigation solution and the output from the EKF (see Figure 1.3). The output of the EKF is the estimate of the errors in the INS navigation solution. Therefore, the plane is actually flying on the INS output corrected by the EKF's best estimate of the errors committed by the INS. The EKF bases its estimate on information it receives from the INS, GPS, and RRS. Therefore, the aircraft is navigated using the information provided by all three navigation systems. A more complete description of the EKF and how it is used in the NRS is provided in Chapters II and III.

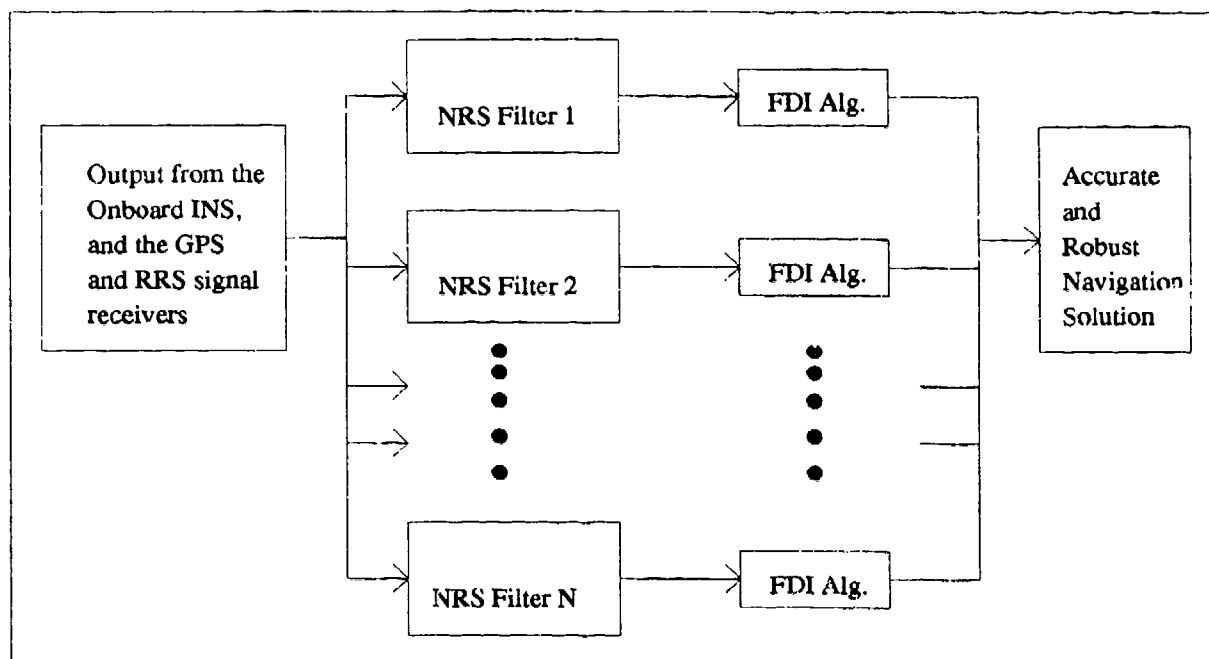


Figure 1.4 MNRS System Description

The MNRS is basically a multiple model implementation of the original NRS. The MNRS consists of a bank of NRS navigating filters which run in parallel (see Figure 1.4). Each NRS filter produces its own navigation solution. Unlike traditional multiple model schemes, each NRS filter produces an accurate solution when no measurement failures occur. The NRS filters have been designed in such a way that a failure in one measurement signal (GPS or RRS) will affect all but one of the NRS filters. Whether a failure has occurred or not is decided in a failure detection test conducted on each of the NRS filters (see Figure 1.4). When a failure occurs, all the NRS filters but one will fail the detection test, and navigation of the aircraft then continues on the single filter that passes its test. This allows for robust navigation during the failure. A more complete description of this FDIR algorithm is provided in the literature review, Section 1.6. The real benefit of using the MNRS is that during single failures, one and only one of the NRS filters will remain unaffected, which allows for highly accurate navigation during measurement signal failures.

1.5 Assumptions

All theses are limited by the assumptions made, and no research can be adequately evaluated unless these assumptions are clearly defined. This section outlines the assumptions that have been made in this thesis.

1. All work has been conducted through computer simulation. The "real" world in the simulation is modelled as a full-order-state truth model. The MNRS filter is represented by a bank of reduced-order filter models. This modelling structure has been chosen to validate the function of the EKF's in the MNRS [9]. Once this computer simulation work has been completed and verified, a decision can be made as to whether it is beneficial to put this theory into application. Yet this decision cannot be made until complete and thorough simulation work has been completed. The full-order truth and filter models are presented in Chapter III.
2. The INS platform is assumed to be stabilized with a barometric altimeter. An INS platform is unstable without an outside measurement source in the vertical channel [1]. While a barometric altimeter is not the only way to stabilize a platform, it is a commonly used method. The use of the barometric altimeter is included in the modelling of the system.
3. A sample period of two seconds has been chosen for the EKF. The sample period refers to how often GPS and RRS error measurements will be brought into the EKF. Past AFIT research has used a variety of sample periods, varying from two to ten seconds for the NRS. The decision to use the two second sample period has been based primarily on the computer facilities available and the accuracy of navigation solution produced. Various runs at different sample periods confirm that this sample period is valid for simulation purposes [19].
4. The MNRS model used consists of a bank of NRS models in parallel. Each of these models has been developed using differential equations representing the error characteristics of the navigational system operating in the real world. The models are based on the theoretical work conducted at CIGTF and AFIT in developing error state models for the GPS, RRS, and INS navigational systems. The GPS and RRS

models are generic models [14], while the INS model is based on the Litton LN-93 strap-down inertial navigation system [4]. A more complete description of the system models is presented in Chapter III.

5. Failures are assumed to be additive in nature rather than multiplicative. This assumption has been made because it both greatly simplifies the math and it describes most sensor failures that occur in the real world. The algorithms presented are valid only for failures of an additive nature.
6. The computer simulations have been developed using a program called Multi-mode Simulation for Optimal Filter Evaluation (MSOFE) [13]. MSOFE is established Air Force software to develop and test Kalman filter algorithms. The computer-simulated flight profile has been generated by the program PROFGEN [12]. PROFGEN is designed to work with MSOFE to provide the necessary data files to simulate dynamic flight profiles. The FDIR algorithms are implemented using the commercial package software MATLAB [8].
7. The state dynamics matrix \mathbf{F} is considered piecewise constant, i.e. constant between sample times two seconds apart. This assumption is necessary to implement the discrete time GLR matching filters algorithm developed in Section 2.3.2.2. The validity of this assumption has been documented in past AFIT research [19].
8. The simulation software, MSOFE and MATLAB, has been coded to run in double precision to increase numerical stability of the simulation. This is necessary due to large disparities in filter variables that need to be operated on during the simulation. Due to the fact that rescaling of the disparate variables is not considered an effective solution for this application, the MSOFE software implements a U-D factorization algorithm to increase the numerical stability in the EKF equations [13, 9].
9. The MSOFE runs are conducted using 15-run Monte Carlo analyses. While a larger batch size for the Monte Carlo analysis would be preferable, this value has been chosen to keep the computational burden of the thesis within reasonable bounds. Single-run Monte Carlo analysis is used for the FDIR runs, to reveal what on-line operational filters would see and do. The single runs are compared to the 15-run

Monte Carlo results to ensure that the single runs are reasonable examples of system operation.

10. The failure thresholds established for the FDIR algorithm have been established empirically, using good engineering insight and verified through exhaustive testing.
11. Taylor series approximations truncated at first order are used for linearizing nonlinear equations in the MNRS filter. Perturbations about a nominal trajectory will be established in each case.
12. A Doppler system is used to provide velocity aiding to the INS. The measurements from the Doppler are assumed to be ideal and tell the filter the exact error between the filter state and the truth state. This assumption is relatively poor, yet the development of a complete Doppler error model is beyond the scope of this thesis. Also the use of the ideal Doppler measurement allows for direct comparison to past AFIT research, [14, 19].

1.6 Literature Review

The goal of this literature review is to provide background information leading up to the choice of FDIR algorithm implemented in the integrated MNRS. The review begins with an introduction to the basic concepts of failure detection, isolation, and recovery (FDIR). Next, various techniques will be analyzed with the criterion of applicability to the navigation model. Once a background in failure analysis techniques has been established, a combination of techniques will be presented in the form of the MNRS algorithm. The literature review will conclude with an argument for the use of the MNRS FDIR technique for accurate and robust navigation.

1.6.1 The Concept of Failure Analysis. Failure detection algorithms analyze multiple information sources to determine if erroneous information has entered a system. The erroneous information could be anything from a bias on an incoming measurement to a hard failure of an entire subsystem. Each algorithm differs in function and complexity. Simple algorithms simply detect the existence of failures in the system. More complicated algorithms actually isolate and recover from failures. However, despite the differences in

various algorithms, the fundamental objective of failure analysis is to examine information in such a way that failures can be seen as discrepancies between different information sources [21:pp. 601].

Failure analysis hinges on the availability of multiple information sources providing redundant information. With only one source of information, failed operation cannot be distinguished from normal operation of a system. For example, imagine you are driving a car in which the only source of information about your speed is the speedometer. In this car, the driver has no choice but to rely on the speedometer for his speed information. This imaginary car cannot employ failure analysis, because it contains no redundant information sources. Yet in a real car, failure analysis algorithms are constantly employed. In a real car, the driver has access to multiple sources of information about his speed; the speedometer, the sound of the engine, visual cues such as the blurring of the scenery, and the tone of his passenger's voice. If the information from the speedometer is contrary to all his other information sources, the wise driver notes the inconsistency, slows the vehicle, and declares a failure in his speedometer. The driver has just performed the function of FDIR. While the above example may seem trivial, all FDIR algorithms perform the same function as the driver did analyzing his multiple information sources.

It is important to note in the example that, while all the sources of information are providing data about the speed of the car, all the information is formatted in different reference frames (speedometer in miles/hr, the passenger in decibels). An FDIR algorithm needs to transform all the information into one frame, so that they can be compared equitably. If the information is not transformed to a common frame, there is no basis for comparison [6:pp. 400]. This concept will reappear in the development of the FDI algorithm for the integrated NRS in Chapter III.

Another important concept in failure analysis is the identification of failure types. When a measurement or component of a system fails, knowledge of the type of failure that has occurred can be crucial to correcting the problem. For instance, when a speedometer goes bad, the driver can compensate for this if he knows the type of failure that has happened (i.e., is the speedometer too slow or too fast?). If the driver knows the type

of failure, he can adjust the value of the measurement he reads from the speedometer accordingly. This process is called corrective feedback through failure mode analysis.

Now that a basic understanding of failure analysis has been established, a review of various failure analysis techniques is presented. Each technique will be evaluated on its ability to conduct effective FDIR.

1.6.2 The Voting Method. The simplest form of failure analysis is voting. To implement simple voting, one needs at least three redundant information sources. Failure detection and isolation is conducted by looking at the three sources and seeing if one is not in agreement with the other two. If one source of information disagrees with the other two, then the inconsistent source is declared a failed source [21:pp. 604-605]. If it is possible to continue operation with only two information sources, the voting algorithm is able to recover from the failure and continue operation.

The benefit of this method is its utter simplicity. Unfortunately, simplicity is also its major drawback. This FDIR algorithm assumes majority rule. While this is very democratic, it does nothing for the user if the one source happens to be right while the other two are wrong. This dilemma can be cured by increasing the number of voters. Yet massive sensor redundancy is impractical in the limited space and computational domain of avionics [21:pp. 604-605]. Therefore, the voting algorithm was not seriously considered for the integrated navigation system.

1.6.3 The Chi-Square Test ($\chi(k)$). The Chi-Square testing algorithm[21] assumes that a Kalman filter is being used to blend the multiple information sources. A complete presentation of Kalman filter theory is presented by Maybeck [9]. The Chi-Square test operates on the measurement residuals of the Kalman filter and decides whether a failure has occurred. The Chi-square test is a function of both the measurement residual vector and the filter-computed covariance of the residual vector. The Chi-Square test therefore is a function of both the size of the residuals and the Kalman filter's confidence in the size of the residuals. The Chi-Square test declares a failure when the $\chi(k)$ function is greater than a pre-determined threshold. The threshold is violated when the difference between the

Kalman filter estimate of the incoming measurement and the actual measurement grows large with regard to the filter-predicted covariance of the residual. Past work at AFIT has shown the Chi-Square test to be a highly effective and consistent failure detection technique [2, 5].

While the Chi-Square test provides excellent failure detection, the algorithm is incapable of performing failure isolation or recovery. As a result, the Chi-Square test cannot be classified as a stand-alone FDIR algorithm [21:pp. 606-607].

1.6.4 Generalized Likelihood Ratio Testing (GLR). The GLR test[20], although similar to the Chi-Square test, provides for both detection and isolation of failures. Like the Chi-Square test, the GLR test uses the residuals of a Kalman filter as its basis for failure analysis. Yet the GLR test uses the residuals in such a way that it is able both to detect and to isolate failures [20].

The GLR test feeds the residuals of the Kalman filter into a bank of estimators, each designed to look for a certain type of failure mode (see Figure 1.5). Examples of failure modes are no failure, a step failure, and a ramp failure. Each estimator tests a hypothesis, H_k , corresponding to a possible failure affecting the system. H_0 corresponds to the no-failure hypothesis, while H_1 and H_2 are the bias and the ramp failure hypotheses respectively. Each estimator conducts its own maximum likelihood estimate (MLE) for its specific hypothesis. The results of each MLE are fed into a common test logic algorithm (see Figure 1.5). This algorithm decides which hypothesis is true, thereby determining which, if any, failure has occurred.

One of the key benefits of the GLR test is that it needs only *one* estimator for each type of failure. The GLR algorithm actually estimates unknown variables such as the magnitude of the failure type in its FDI process; therefore only one estimator is needed for each type of failure. This aspect greatly reduces the computational load of the GLR in comparison to other multiple model techniques. Also, the estimate of the failure magnitude can be used to help develop a feedback loop to allow for reconfigurability of the system. While it is *possible* to use the GLR as a complete FDIR technique, past AFIT research has

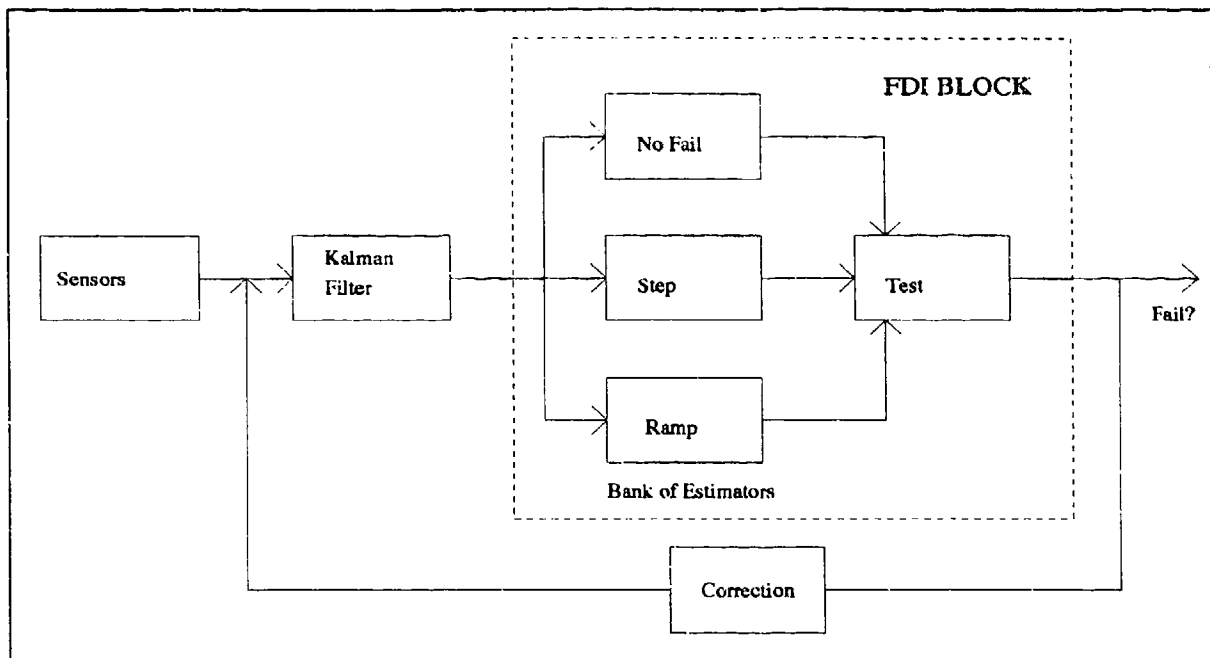


Figure 1.5 Multiple GLR Testing

been unable to produce accurate enough failure estimates to implement effective system recovery [19].

1.6.5 Multiple Model Adaptive Estimation (MMAE). MMAE, like the GLR and the Chi-Square tests, exploits the information provided by the residuals of Kalman filters. The strength of MMAE lies in its rapid reconfigurability. By running multiple filters in parallel, residual information at each update is used to instantly reconfigure the system to failures. Unlike the previous algorithms which implement only a single Kalman filter (see Figure 1.5), the multiple model structure of MMAE is fundamentally designed for robustness to known system failures.

To describe MMAE, first a failure space is defined as all the possible operating modes of a system under all possible failures that are to be considered. The MMAE can be robust to system failures because it implements a bank of Kalman filters that span the failure space of the system (see Figure 1.6). Each Kalman filter in the bank models the operation of the system under a different assumed failure mode. To get a single navigation solution out of the bank of filters, the outputs of all the filters are blended together through a probability-

weighted average based on the probability of that filter modeling what is actually occurring (see Figure 1.6). Therefore, when no failure has occurred, the filter modelling *no failure* gets a very high weight (almost one), while all the other models will get very low weights (almost zero). When a failure occurs those filters with the best model of the failure will get the highest weighting probabilities. In Figure 1.6, one can see that the residuals of each Kalman filter are fed into the algorithm that computes the relative weights placed on each filter estimate [10]. The real benefit of MMAE is that the bank of Kalman filters spans

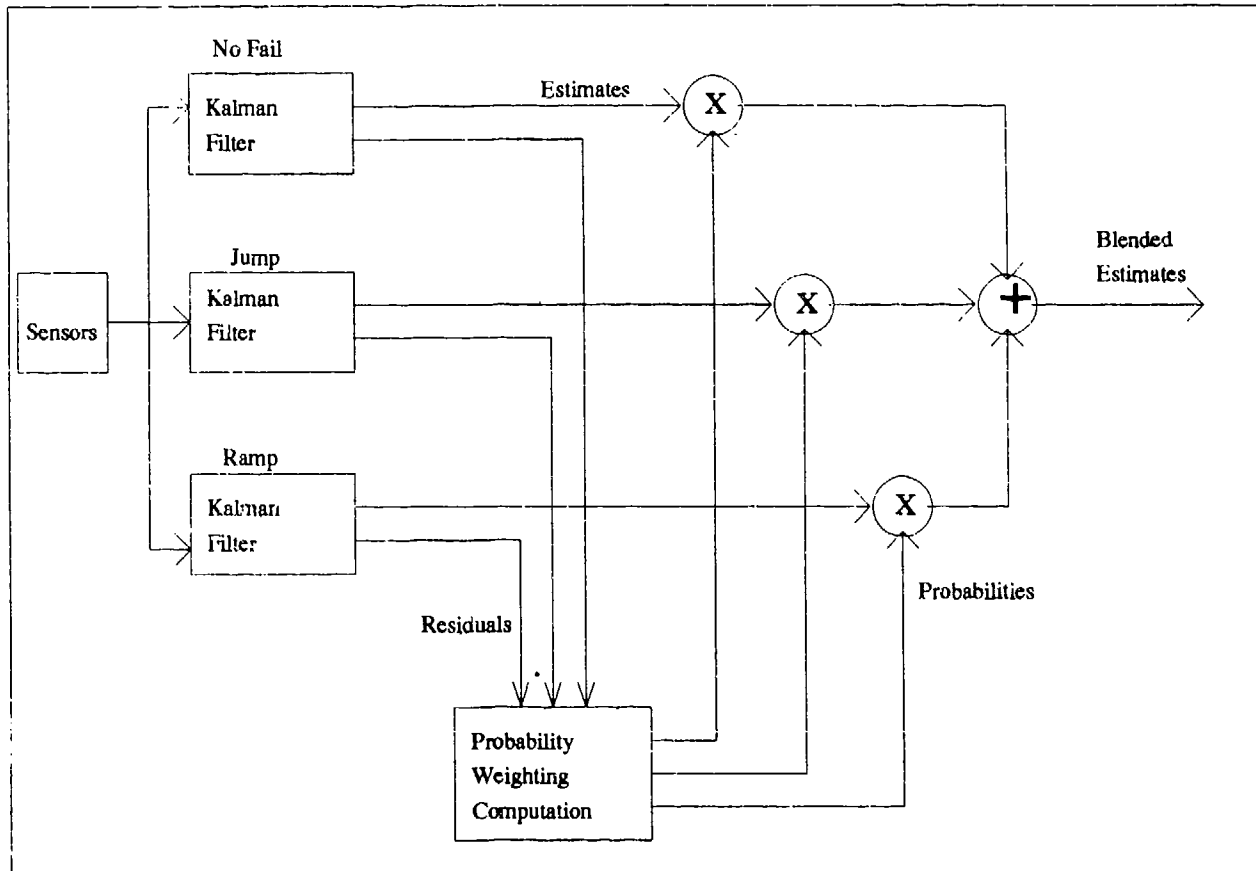


Figure 1.6 Multiple Model Adaptive Estimation

an entire failure space. As long as the system being affected by a failure remains within the span of this field, the filter is robust to the occurrence of that failure [11]. Therefore within the scope of the failure space, MMAE is an extremely effective FDIR algorithm.

The major drawback of MMAE is the high number of Kalman filters that can be necessary to span a realistic failure space. If MMAE is implemented with too few filters,

none of the Kalman filters will have good looking residuals, thus limiting the accuracy of the blended state estimates. Yet the necessary number of models can rapidly grow into the hundreds, when mapping a large failure space. For this reason, a modification of the MMAE algorithm can often be more effective than MMAE in its pure form [21].

1.6.6 A Combination of Techniques. The three FDI algorithms, MMAE, GLR test, and Chi-Square test, each have their benefits and drawbacks. For the MNRS the goal is to construct a single FDIR algorithm that exploits the best of all three methods.

The proposed MNRS algorithm is fundamentally a multiple model FDIR algorithm. Multiple Kalman filters run in parallel, each identical in structure, yet receiving different sets of measurements. Table 1.1 shows the differences in the incoming measurements for the ten filters used in the parallel algorithm. The first filter is the one that will be used under a no-failure situation. It will receive measurements from those satellites and transponders geometrically located to give the best navigation information. While this filter is running, nine other filters are running in parallel. In each of these nine filters, one of the original signals has been switched with an alternate signal (either a new GPS signal or transponder signal). Therefore, none of the ten filters is being updated with the same measurement information. In Table 1.1, satellites 1-4 and transponders 1-5 combine for the optimal navigation solution using the MNRS. Satellite 5 and transponder 6 are the alternate signals which allow for the reconfigurability under failure conditions.

Table 1.1 MNRS Filter Measurement Structure

Filter	Satellites Used	Transponders Used
1	1,2,3,4	1,2,3,4,5
2	1,2,3,5	1,2,3,4,5
3	1,2,4,5	1,2,3,4,5
4	1,3,4,5	1,2,3,4,5
5	2,3,4,5	1,2,3,4,5
6	1,2,3,4	1,2,3,4,6
7	1,2,3,4	1,2,3,6,5
8	1,2,3,4	1,2,6,4,5
9	1,2,3,4	1,6,3,4,5
10	1,2,3,4	6,2,3,4,5

With the above modelling structure, failure detection is conducted with the Chi-Square test. The Chi-Square evaluation is run on all ten filters (see Figure 1.7). If one of satellites 1-4 or one of transponders 1-5 fail, the Chi-Square algorithms will signal a failure in nine of the ten filters (including Kalman filter 1, the navigating filter). At this point the navigation solution will be switched from Kalman filter 1 to whatever filter has not failed (see Figure 1.7). In this way the MNRS is robust in the face of single failures in satellite and transponder signals.

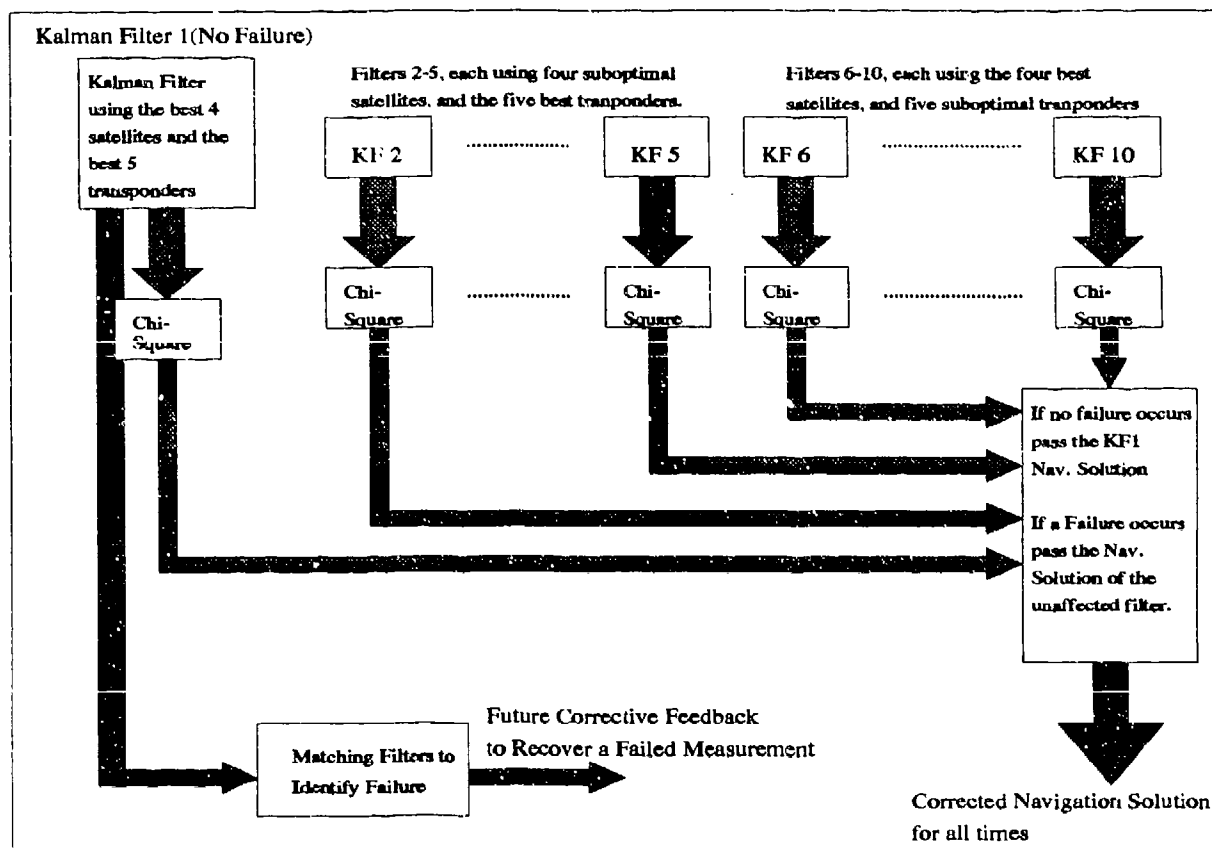


Figure 1.7 Combined FDI Algorithm

A major difference between this algorithm and normal MMAE lies in the fact that, during the no failure case, all the separate filters are producing an accurate solution. In a normal MMAE, only one accurate solution is ever produced. Since only one navigation solution is needed in the no-failure case, the MNRS chooses the best filter output based on the location of the RRS transponder sites and the GPS satellites. Therefore the MNRS is

fundamentally a combination of the original MMAE technique modified to exploit existence of redundant sensors (voting technique) and the high quality of the Chi-Square tests.

The MNRS does not use any type of feedback to correct the failed measurement. The failed measurement is abandoned. While this removal of the failed measurement from the system allows for highly accurate results, information is lost when the measurement is abandoned. Therefore, an effort is made to examine matching filter techniques to identify the type of failure and estimate its magnitude. Two matching filter algorithms are considered. The first matching filter algorithm is derived from the GLR test. This filter attempts to distinguish between step and ramp failures. A failure magnitude estimator can also be derived from the GLR algorithm. The second matching filter algorithm is an ad-hoc design based on recognizing failure types in the output of the Chi-Square test. In the MNRS diagram, Figure 1.7, these two different matching filter algorithms are represented by the matching filter block. It is hoped that this failure identification work will allow recovery of failed measurements in the future.

The structure for this model has been inspired by two separate sources; multiple model adaptive estimation presented by Maybeck [11] and a robust navigation system proposed by Schwartz [16]. By using different aspects of both studies, the above algorithm has been formulated. It should be noted that the MNRS algorithm can detect up to two failures of the system, yet it can only isolate and reconfigure for one failure at a time.

1.6.7 Literature Review Conclusion. The Chi-Square test, MMAE algorithms, and matching filters are combined to produce the MNRS FDI algorithm. Each of the individual failure analysis techniques have characteristics that are exploited for the benefit of the overall navigation system. MMAE provides the concepts of instant reconfigurability and multiple modelling to the combined FDI algorithm. The Chi-Square test is exploited for its rapid failure detection. Using the Chi-Square in each of the distributed filters allows for the validity of a filter to be declared as fast as possible. Finally, the matching filters are used to identify the type of failure. In this way, future corrective feedback can be used to bring the failed models back on line with the failure compensated.

1.7 Plan of Attack

The work for this thesis divides itself into three distinct tasks to be completed. These three steps in the research are presented in Sections 1.7.1, 1.7.2, and 1.7.3. The first two steps encompass the work needed to get the basic MNRS operational, while the third step looks at ways of expanding the MNRS for future research.

1.7.1 Preliminary Research. The introductory research began with the work leading up to the choice of the MNRS for robust navigation. The literature review, presented in Section 1.6, is the culmination of this research.

A preliminary study was made to see if a multiple model system could adequately detect single failures. This preliminary test was conducted as a class project for the AFIT EE735 class. The project used a simplified INS/GPS model and employed a multiple model structure similar to the MNRS approach. The multiple filter system designed by the EE735 class looked for step, ramp, and noise offsets on incoming GPS signals. The project was considered only a preliminary study because simplified filter and truth models were used and because transponder failures were not considered. The completed report for this project demonstrated the effectiveness of the multiple model approach for robust navigation [2].

The preliminary studies concluded when a choice was made to go with MNRS system using the multiple model approach with the Chi-square test as the failure detection and isolation algorithm. Once the groundwork of the thesis had been established, the actual system development and testing could begin.

1.7.2 System Development and Testing. The second phase of research is presented as a three-step process beginning with model development and ending with FDIR verification.

1. Rework the original INS Litton LN-93 truth and filter models based on the most recent research completed by the Avionics Directorate, System Avionics Division, Wright Laboratory (WL/AAAS), AFMC.

2. Integrate the new INS model with the original GPS and RRS models, and recode all the models on the Unix-based SPARC computer system. Verify the operation of the filter model against the truth model.
3. Develop the computer code to create the MNRS system. Test this multiple model structure for correct operation in a single-failure environment.

1.7.3 Advanced Research into Adaptive Techniques. The final goal for this research was to compare two matching filter algorithms to determine the performance capabilities of each. While the effectiveness of the matching filters is not crucial to the operation of the MNRS, this work can lead to future implementation of corrective feedback algorithms for the MNRS. The implementation of corrective feedback utilizing the matching filters is beyond the scope of this thesis.

1.8 Overview of Thesis

Chapter II presents the detailed theory used in the research. Kalman filter theory is introduced with emphasis on the development of the extended Kalman filter. The theory behind the various failure analysis algorithms is also explained, including the equations implemented for the Chi-square and the matching filter algorithms. Finally, the analysis used for selecting thresholds is discussed.

Chapter III fully describes the navigation system's parameters and operational details through an overall system description. Models for the INS, GPS and RRS portions of the NRS model are defined in detail. A description of how the NRS filters are used in the MNRS is also presented. This Chapter also presents various failure models implemented in the thesis.

Results of the work accomplished are shown in Chapter IV. The reduced order extended Kalman filter used in the MNRS filters is analyzed, and a discussion of the FDIR performance is presented. Chapter V summarizes the research with conclusions and recommendations.

II. Kalman Filtering and Failure Detection

2.1 Overview

This section presents the fundamental theory for the application of Kalman filter theory to the navigation problem. First, the necessary update and propagation equations are developed for a sampled-data extended Kalman filter (EKF). The EKF theory supports the basic operation of the NRS model. Next, the failure detection and isolation theory implemented in the MNRS is presented. This theory focuses on the development of the Chi-Square test and two matching filter algorithms. The chapter concludes with a discussion of the threshold selection criteria for failure declaration algorithms.

2.2 The Extended Kalman Filter

The MNRS model is generated using error state, extended Kalman filter (EKF) equations. The EKF provides excellent state estimation of the non-linear time-varying stochastic navigation equations. These stochastic equations are linearized through approximation techniques about a nominal trajectory to form a linearized Kalman filter (LKF). The LKF is the basis for the EKF. The EKF expands upon the LKF by allowing the nominal trajectory to vary over time. The EKF relinearizes its dynamics and measurement equations about the best state estimate about each update from the GPS and the RRS measurements. The EKF that is implemented in this work is considered a sampled data Kalman filter since the measurement updates enter the system at discrete times.

2.2.1 The Sampled Data Kalman Filter. Let the system model be expressed as a state equation of the form

$$\dot{x}(t) = f[x(t), t] + G(t)w(t) \quad (2.1)$$

where the state dynamics vector $f[x(t), t]$ is a nonlinear function of the state vector $x(t)$ and time t . Let the process noise input matrix $G(t) = I$ and $w(t)$ be a white Gaussian noise with mean:

$$m_w = E\{w(t)\} = 0 \quad (2.2)$$

and noise strength $Q(t)$ defined by:

$$E \{ w(t)w^T(t + \tau) \} = Q(t)\delta(\tau) \quad (2.3)$$

Incorporate measurements $z(t_i)$ into the filter at discrete times and define them as a non-linear function of the state vector and time:

$$z(t_i) = h[x(t_i), t_i] + v(t_i) \quad (2.4)$$

where $v(t_i)$ is a zero-mean white Gaussian noise of covariance $R(t_i)$ defined by:

$$E \{ v(t_i)v^T(t_j) \} = \begin{cases} R(t_i) & \text{for } t_i = t_j \\ 0 & \text{for } t_i \neq t_j \end{cases} \quad (2.5)$$

and $h[x(t_i), t_i]$ is the nonlinear observation vector. The LKF is based on *perturbation* states about a nominal state trajectory $\mathbf{x}_n(t)$ satisfying $\mathbf{x}_n(t_0) = \mathbf{x}_{n_0}$ and

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), t] \quad (2.6)$$

where $f[\cdot, \cdot]$ is shown in Equation (2.1). The measurements are also based on the nominal states and defined as:

$$z_n(t_i) = \mathbf{h}[\mathbf{x}_n(t_i), t_i] \quad (2.7)$$

The perturbation states are found by subtracting the nominal states in Equation (2.6) from the original states in Equation (2.1):

$$[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)] = \mathbf{f}[\mathbf{x}(t), t] - \mathbf{f}[\mathbf{x}_n(t), t] + G(t)w(t) \quad (2.8)$$

Equation (2.8) is approximated to first order through a truncated Taylor series expansion:

$$\delta \dot{\mathbf{x}}(t) = \mathbf{F}[t; \mathbf{x}_n(t)] \delta \mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \quad (2.9)$$

where $\delta x(t)$ are the perturbation states. The definitions for $G(t)$ and $w(t)$ are unchanged and the new linearized dynamics matrix $F[t; \mathbf{x}_n(t)]$ is found by taking partial derivatives of $\mathbf{f}[\mathbf{x}(t), t]$ with respect to $\mathbf{x}(t)$ and evaluated at the nominal values for the trajectory $\mathbf{x}_n(t)$:

$$\mathbf{F}[t; \mathbf{x}_n(t)] = \left. \frac{\partial \mathbf{f}[\mathbf{x}(t), t]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_n(t)} \quad (2.10)$$

The discrete-time measurements are similarly approximated to first order and are in the perturbed form:

$$\delta z(t_i) = \mathbf{H}[t_i; \mathbf{x}(t_i)] \delta \mathbf{x}(t_i) + \mathbf{v}(t_i) \quad (2.11)$$

and the same linearization process is used for the measurement matrix $H[t_i; \mathbf{x}_n(t_i)]$, resulting in:

$$\mathbf{H}[t_i; \mathbf{x}_n(t_i)] = \left. \frac{\partial \mathbf{h}[\mathbf{x}(t_i), t_i]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_n(t_i)} \quad (2.12)$$

The LKF in this thesis generates error state estimates $\widehat{\delta \mathbf{x}}(t)$, which can be added to the nominal states to provide whole states estimates $\hat{\mathbf{x}}(t)$ in the form:

$$\hat{\mathbf{x}}(t) = \mathbf{x}_n(t) + \widehat{\delta \mathbf{x}}(t) \quad (2.13)$$

The EKF will now be formed by linearizing about the state estimate $\hat{\mathbf{x}}$ rather than the nominal trajectory \mathbf{x}_n . The following equations use the notation t/t_i to represent the value of a variable at time t , conditioned on the measurements taken through the time t_i , for t being any time in the time interval $[t_i, t_{i+1})$. Also, t_i^- represents the value after propagation but prior to the measurement update and t_i^+ corresponds to the value after the measurement update. The state estimates and covariance values $P(t/t_i)$ are propagated from t_i to t_{i+1} by solving the following differential equations:

$$\dot{\hat{\mathbf{x}}}(t/t_i) = \mathbf{f}[\hat{\mathbf{x}}(t/t_i), t] \quad (2.14)$$

$$\dot{P}(t/t_i) = \mathbf{F}[t; \hat{\mathbf{x}}(t/t_i)]P(t/t_i) + P(t/t_i)\mathbf{F}^T[t; \hat{\mathbf{x}}(t/t_i)] + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \quad (2.15)$$

where

$$F[t; \hat{x}(t/t_i)] = \left. \frac{\partial f[x(t), t]}{\partial x} \right|_{x=\hat{x}(t/t_i)} \quad (2.16)$$

and initial conditions are given by:

$$\hat{x}(t_i/t_i) = \hat{x}(t_i^+) \quad (2.17)$$

$$P(t_i/t_i) = P(t_i^+) \quad (2.18)$$

The discrete-time measurements are processed in the EKF through update equations:

$$K(t_i) = \mathbf{P}(t_i^-) \mathbf{H}^T[t_i; \hat{x}(t_i^-)] \left\{ H[t_i; \hat{x}(t_i^-)] \mathbf{P}(t_i^-) \mathbf{H}^T[t_i; \hat{x}(t_i^-)] + R(t_i) \right\}^{-1} \quad (2.19)$$

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i) \{z_i - h[\hat{x}(t_i^-), t_i]\} \quad (2.20)$$

$$P(t_i^+) = P(t_i^-) - K(t_i) H[t_i; \hat{x}(t_i^-)] P(t_i^-) \quad (2.21)$$

where

$$H(t_i) = H[t_i; \hat{x}(t_i^-)] = \left. \frac{\partial h[x(t_i), t_i]}{\partial x} \right|_{x=\hat{x}(t_i^-)} \quad (2.22)$$

and $K(t_i)$ is the discrete-time Kalman filter gain. Note that, for the EKF, the measurement and dynamics matrices are relinearized about the last state estimate $\hat{x}(t_i^-)$ rather than the nominal trajectory used by a simple linearized Kalman filter.

2.2.2 The Discrete-Time Kalman Filter. In order to utilize the filter outputs in the GLR matching filter algorithm, it is necessary to discretize the state dynamics matrix into a state transition matrix (STM), $\Phi(t_{i+1}, t_i)$. All other quantities of interest such as K and H are already in discrete form. The STM must satisfy the following differential equation and initial condition [9]:

$$d[\Phi(t, t_i)]/dt = F(t)\Phi(t, t_i) \quad (2.23)$$

$$\Phi(t_i, t_i) = I \quad (2.24)$$

Defining $\Delta t = t_{i+1} - t_i$ and solving with F assumed constant over Δt (see Assumption 7)) leads to:

$$\Phi(t_{i+1}, t_i) = e^{F\Delta t} \quad (2.25)$$

The state equation, Equation (2.9) can now be written in the discrete form

$$\delta x(t_{i+1}) = \Phi(t_{i+1}, t_i)\delta x(t_i) + \mathbf{G}_d(t_i)\mathbf{w}_d(t_i) \quad (2.26)$$

where \mathbf{G}_d is assumed to be the identity matrix and \mathbf{w}_d is defined as a discrete-time zero-mean white Gaussian noise sequence with covariance kernel:

$$E\{w_d(t_i)w_d^T(t_j)\} = \begin{cases} \mathbf{Q}_d(t_i) & \text{for } t_i = t_j \\ 0 & \text{for } t_i \neq t_j \end{cases} \quad (2.27)$$

2.3 Failure Detection Algorithms

This section presents the theory for three failure analysis algorithms: the Chi-Square test, the GLR matching filter, and the Chi-Square residual matching filter. The Chi-Square test is the failure detection algorithm for each NRS filter in the MNRS model. The two matching filter algorithms are candidates for the failure type identification algorithm (identifying between failures). For an explanation as to how these algorithms are combined in the MNRS model, see Sections 1.6.

2.3.1 Chi-Square Equations. The Chi-Square test is designed to detect anomalies in the residuals of a Kalman filter. When the magnitude of the residuals remains larger than anticipated through the filter-computed residual covariance matrix over a window of time, the Chi-Square test declares a failure. The Chi-Square test is a highly simplistic yet powerful tool that is capable of detecting failures rapidly and accurately.

The Chi-Square test declares a failure based on the size of the the extended Kalman filter residuals, $\gamma(t_i)$. The EKF residuals, first seen in Equation (2.20), are defined in Equation (2.28). These residuals are zero mean and appear white to first order with known covariance, $\Lambda(t_i)$.

$$\gamma(t_i) = z(t_i) - h[\hat{x}(t_i^-), t_i] \quad (2.28)$$

$$\Lambda(t_i) = H(t_i)P(t_i^-)H^T(t_i) + R(t_i) \quad (2.29)$$

The Chi-Square test is a function of the Chi-Square random variable, $\chi(t_k)$, and is given by

$$\chi(t_k) = \sum_{i=k-N+1}^k \gamma^T(t_i)\Lambda^{-1}(t_i)\gamma(t_i) \quad (2.30)$$

with N being the size of a sliding window in time. In Equation (2.30), the residuals, $\gamma(t_i)$, are squared to remove sign cancellation in the summation. The square of the residuals is scaled by the inverse residual covariance, to take into account the confidence the filter has in its residuals at that moment. The use of the sliding window in time, rather than a full history from t_0 , reduces the computational load and increases the accuracy of the detection algorithm. The delay time from failure occurrence to failure detection is a function of the size of the window, N . A small window size decreases the detection time, however the small window also increases the likelihood of false alarms. Therefore the size of the window, N , is chosen to balance the trade-off between detection delay and false alarm rate. The Chi-square test will declare a failure based on a simple threshold test,

$$\begin{aligned} \chi(t_k) &> \epsilon \Rightarrow \text{FAILURE} \\ \chi(t_k) &\leq \epsilon \Rightarrow \text{NO FAILURE} \end{aligned} \quad (2.31)$$

The end of Chapter II will present more detail on the methodology of threshold choosing, and Chapter III will go into greater detail as to how the Chi-Square algorithm is implemented in the MNRS model.

2.3.2 GLR Matching Filter. The GLR matching filter algorithm is designed both to distinguish between different types of failures and to estimate the magnitudes of the failure types. The GLR matching filter algorithm pursued in this thesis is derived from the generalized likelihood ratio equations [20]. Therefore, before the GLR matching filters are

presented, the generalized likelihood ratio will be fully presented. Afterwards, the GLR matching filters used in the algorithm will be derived from the fundamental GLR theory.

2.3.2.1 Derivation of the GLR Equations. The derivation of the GLR test will be divided into five steps. The derivation centers around the development of a likelihood equation based on two different hypotheses. One hypothesis, H_0 , represents a no-failure system. The second hypothesis, H_1 , represents a specific failure type added to the system. As stated earlier in Section 1.3, the failures induced for this thesis will be additive terms in the measurement updates. The ratio of the log-likelihood of the two hypotheses will be used to generate a threshold function, $l(t_i, \theta)$, which satisfies the following criteria,

$$\begin{aligned} l(t_i, \theta) &> T \Rightarrow \text{FAILURE} \\ l(t_i, \theta) &\leq T \Rightarrow \text{NO FAILURE} \end{aligned} \quad (2.32)$$

If $l(t_i, \theta)$ is less than some threshold, T , then H_0 is true. Similarly H_1 is true if $l(t_i, \theta)$ is greater than the T . The following steps describe the derivation of the GLR algorithm for a single failure type, a step failure.

1. Step One, Define the Hypotheses H_0 and H_1

The derivation of the GLR test begins with the establishment of the two hypotheses, each based on a different set of discretized extended Kalman filter equations. The first hypothesis, H_0 (no failure), comes from discretized EKF equations (Equations (2.33) and (2.34)) that assume that no failure has occurred.

$$\delta x(t_{i+1}) = \Phi(t_{i+1}, t_i) \delta x(t_i) + \mathbf{G}_d(t_i) \mathbf{w}_d(t_i) \quad (2.33)$$

$$\delta z^0(t_i) = \mathbf{H}(t_i) \delta x(t_i) + \mathbf{v}(t_i) \quad (2.34)$$

The second hypothesis assumes that an additive failure has occurred on the measurement signal. Throughout this section the superscript notation, 0 and 1 refer to variables associated with Hypotheses H_0 and H_1 respectively. These equations have

an extra term that reflects the nature of the failure (Equations (2.35) and (2.36)).

$$\delta x(t_{i+1}) = \Phi(t_{i+1}, t_i) \delta x(t_i) + \mathbf{G}_d(t_i) \mathbf{w}_d(t_i) \quad (2.35)$$

$$\delta z^1(t_i) = \mathbf{H}(t_i) \delta x(t_i) + \mathbf{v}(t_i) + d(t_i) n(t_i, \theta) \nu \quad (2.36)$$

where

$d(t_i)$	=	failure vector
$n(t_i, \theta)$	=	failure function
ν	=	unknown size of the failure
θ	=	unknown time of the failure

Comparison of Equations (2.34) and (2.36) reveals that the H_1 hypothesis is characterized by the failure offset term, $d(t_i) n(t_i, \theta) \nu$. The components of this term determine the type of failure that the GLR algorithm is trying to detect. The $n(t_i, \theta)$ dictates the time of failure onset, θ , and the type of failure that has occurred, i.e. ramp offset, step offset, etc. The ν is the magnitude of the failure. The column vector, $d(t_i)$, specifies which of the measurement signals has the failure. Chapter III will fully describe the failure models for both $d(t_i)$ and $n(t_i, \theta)$. One of the real benefits of the GLR algorithm, as stated in Section 1.6, is that the unknown variable, ν , is estimated in the GLR algorithm. Therefore ν does not need to be predetermined for the hypothesis, H_1 .

2. Step Two Development of the Residual Equations

Similar to the Chi-Square algorithm, the GLR test focuses on the residuals for its failure information. Therefore the GLR derivation hinges on the development of residual equations for the two different hypotheses H_0 and H_1 . These residual equations are derived from Kalman filter equations developed for each of the two hypotheses. Equations (2.37) and (2.38) represent the residual equations for H_0 and H_1 respectively:

$$\gamma^0(t_i) = \mathbf{H}(t_i) \delta x^0(t_i) + \mathbf{v}(t_i) - \mathbf{H}(t_i) \delta \hat{x}^{0-}(t_i) \quad (2.37)$$

$$\gamma^1(t_i) = \mathbf{H}(t_i) \delta x^1(t_i) + \mathbf{v}(t_i) - \mathbf{H}(t_i) \delta \hat{x}^{1-}(t_i) + d(t_i) n(t_i, \theta) \nu \quad (2.38)$$

Equations (2.37) and (2.38) are rewritten using the following definitions,

$$e^0(t_i) = \delta x^0(t_i) - \delta \hat{x}^{0-}(t_i) \quad (2.39)$$

$$e^1(t_i) = \delta x^1(t_i) - \delta \hat{x}^{1-}(t_i) \quad (2.40)$$

to yield,

$$\gamma^0(t_i) = \mathbf{H}(t_i)e^0(t_i) + \mathbf{v}(t_i) \quad (2.41)$$

$$\gamma^1(t_i) = \mathbf{H}(t_i)e^1(t_i) + \mathbf{v}(t_i) + d(t_i)n(t_i, \theta)\nu \quad (2.42)$$

With these definitions in hand, the next step in the derivation is to find an expression for $e(t_i)$ in terms of known model parameters.

3. Step Three, Define expressions for $e^0(t_i)$ and $e^1(t_i)$

The method for developing the expressions for either $e(t_i)$ will be demonstrated by developing $e^1(t_i)$. Using the expression developed for $e^1(t_i)$ in Equation (2.40), a new expression can be developed by using Equations (2.35), (2.28) and (2.20). This new expression, Equation (2.43) puts $e^1(t_i)$ into a recursive form:

$$e^1(t_{i+1}) = \Phi(t_{i+1}, t_i)e^1(t_i) + \mathbf{G}_d(t_i)\mathbf{w}_d(t_i) - \Phi(t_{i+1}, t_i)K(t_i)\gamma^1(t_i) \quad (2.43)$$

In the above equation, $K(t_i)$ represents the Kalman filter gain at time t_i as defined in Equation (2.19). Equation (2.43) can be further expanded by substituting Equation (2.42) for $\gamma^1(t_i)$. After several algebraic simplifications, Equation (2.43) is rewritten as,

$$\begin{aligned} e^1(t_{i+1}) = & \Phi(t_{i+1}, t_i)[I - K(t_i)\mathbf{H}(t_i)]e^1(t_i) + \mathbf{G}_d(t_i)\mathbf{w}_d(t_i) - \\ & \Phi(t_{i+1}, t_i)K(t_i)[\mathbf{v}(t_i) + d(t_i)n(t_i, \theta)\nu] \end{aligned} \quad (2.44)$$

This is the desired form for $e^1(t_i)$. A similar derivation yields the following expression for $e^0(t_i)$,

$$e^0(t_{i+1}) = \Phi(t_{i+1}, t_i)[I - K(t_i)\mathbf{H}(t_i)]e^0(t_i) + \mathbf{G}_d(t_i)\mathbf{w}_d(t_i) -$$

$$\Phi(t_{i+1}, t_i) K(t_i) \mathbf{v}(t_i) \quad (2.45)$$

The two equations derived in Step 3, Equations (2.44) and (2.45), will be used later in the derivation to attain the final form of $l(t_i, \theta)$.

4. Step Four Derive a new expression for $\gamma^1(t_i)$

This is the crucial step in the derivation. The goal is to develop a new expression for $\gamma^1(t_i)$ in terms of $\gamma^0(t_i)$ and other terms that are readily available from a Kalman filter. To accomplish this we start by rewriting Equation (2.38) in the following form,

$$\gamma^1(t_i) = \gamma^0(t_i) + g(t_i, \theta) \nu \quad (2.46)$$

where $g(t_i, \theta)$ is a function of only parameters available from the EKF. Let $g(t_i, \theta)$ be of the form,

$$g(t_i, \theta) = \mathbf{H}(t_i) f(t_i, \theta) + d(t_i) n(t_i, \theta) \quad (2.47)$$

It is therefore necessary to find an expression for $f(t_i, \theta)$ that will satisfy the above equations. This expression is found by substituting in all the other known quantities into Equation (2.46). Expressions for $\gamma^1(t_i)$ and $\gamma^0(t_i)$ are found in Equations (2.41) and (2.42):

$$\begin{aligned} \mathbf{H}(t_i) e^1(t_i) + \mathbf{v}(t_i) + \\ d(t_i) n(t_i, \theta) \nu &= \mathbf{H}(t_i) e^0(t_i) + \mathbf{v}(t_i) + \\ &[\mathbf{H}(t_i) f(t_i, \theta) + d(t_i) n(t_i, \theta)] \nu \end{aligned} \quad (2.48)$$

After cancellation Equation (2.48) can be rewritten in the form,

$$f(t_i, \theta) = \frac{1}{\nu} [e^1(t_i) - e^0(t_i)] \quad (2.49)$$

It is now beneficial to put Equation (2.49) into a recursive form for $f(t_i, \theta)$. Restating Equation (2.49) for $t = t_{i+1}$ yields the following expression,

$$f(t_{i+1}, \theta) = \frac{1}{\nu} [e^1(t_{i+1}) - e^0(t_{i+1})] \quad (2.50)$$

Now the terms on the right side of Equation (2.50) are expanded so that $f(t_{i+1}, \theta)$ becomes a function of $f(t_i, \theta)$ and other known quantities. Equations (2.44) and (2.45) from Step Four provide expressions for $e^1(t_{i+1})$ and $e^0(t_{i+1})$.

$$\begin{aligned} f(t_{i+1}, \theta) = & \frac{1}{\nu} \{ [\Phi(t_{i+1}, t_i) [I - K(t_i) \mathbf{H}(t_i)] e^1(t_i) + \mathbf{G}_d(t_i) \mathbf{w}_d(t_i) - \\ & \Phi(t_{i+1}, t_i) K(t_i) [\mathbf{v}(t_i) + d(t_i) n(t_i, \theta) \nu] - \\ & (\Phi(t_{i+1}, t_i) [I - K(t_i) \mathbf{H}(t_i)] e^0(t_i) + \mathbf{G}_d(t_i) \mathbf{w}_d(t_i) - \\ & \Phi(t_{i+1}, t_i) K(t_i) \mathbf{v}(t_i)) \} \end{aligned} \quad (2.51)$$

This expression can be simplified to yield,

$$\begin{aligned} f(t_{i+1}, \theta) = & \frac{1}{\nu} \{ \Phi(t_{i+1}, t_i) [I - K(t_i) \mathbf{H}(t_i)] (e^1(t_i) - e^0(t_i)) - \\ & \Phi(t_{i+1}, t_i) K(t_i) d(t_i) n(t_i, \theta) \nu \} \end{aligned} \quad (2.52)$$

Now Equation (2.49) is used to complete the recursive formula for $f(t_{i+1}, \theta)$.

$$f(t_{i+1}, \theta) = \Phi(t_{i+1}, t_i) [I - K(t_i) \mathbf{H}(t_i)] f(t_i, \theta) - \Phi(t_{i+1}, t_i) K(t_i) d(t_i) n(t_i, \theta) \quad (2.53)$$

Therefore at this point, recursive equations have been developed for residuals of the two hypotheses, no failure and failure:

$$H_0 : \gamma(t_i) = \gamma^0(t_i) \quad (2.54)$$

$$H_1 : \gamma(t_i) = \gamma^0(t_i) + g(t_i, \theta) \nu \quad (2.55)$$

The expressions for $g(t_i, \theta)$ can be developed from Equations (2.47) and (2.53).

5. Step Five The Likelihood Ratio

Now that expressions have been developed for the residuals of the two hypothesis, the likelihood ratio needs to be established to determine which of the hypotheses the residuals actually match. We derive the likelihood ratio from the knowledge that both hypotheses residuals are Gaussian random variables that are independent in time [15]. We now define the following probability ratio:

$$L(t_i, \theta, \nu) = \frac{P(\gamma(t_0), \gamma(t_{0+1}), \gamma(t_{0+2}), \dots, \gamma(t_i) | H_1, \theta, \nu)}{P(\gamma(t_0), \gamma(t_{0+1}), \gamma(t_{0+2}), \dots, \gamma(t_i) | H_0)} \quad (2.56)$$

$$L(t_i, \theta, \nu) = \frac{\prod_{t_n=\theta}^{t_i} \left[\frac{1}{\sqrt{2\pi} |\Lambda(t_n)|^{1/2}} e^{-\frac{1}{2}(\gamma(t_n) + g(t_n, \theta)\nu)^T \Lambda(t_n)^{-1} (\gamma(t_n) + g(t_n, \theta)\nu)} \right]}{\prod_{t_n=\theta}^{t_i} \left[\frac{1}{\sqrt{2\pi} |\Lambda(t_n)|^{1/2}} e^{-\frac{1}{2}\gamma(t_n)^T \Lambda(t_n)^{-1} \gamma(t_n)} \right]}$$

The product functions in Equation (2.56) represent the product of the probabilities at each measurement update from time, θ , up to and including the current time, t_i . To simplify Equation (2.56), we use the log-likelihood function in place of the likelihood function. This is accomplished by taking the natural log of both sides of Equation (2.56). After several algebraic simplifications, this equation becomes,

$$\ln(L(t_i, \theta, \nu)) = \nu \sum_{t_n=\theta}^{t_i} [\gamma(t_n)^T \Lambda(t_n)^{-1} g(t_n, \theta)] - \frac{1}{2} \nu^2 \sum_{t_n=\theta}^{t_i} [\gamma(t_n)^T \Lambda(t_n)^{-1} \gamma(t_n)]$$

Similar to the product notation of Equation (2.56), the summation in the equation above represents the sum of the terms at each measurement update from time, θ , up to and including the current time, t_i . This notation continues in Equations (2.58), (2.59), and (2.67). This equation is rewritten in the form,

$$\ln(L(t_i, \theta, \nu)) = \nu \Psi(t_i, \theta) - \frac{1}{2} \nu^2 S(t_i, \theta) \quad (2.57)$$

where $\Psi(t_i, \theta)$ and $S(t_i, \theta)$ are defined below.

$$\Psi(t_i, \theta) = \sum_{t_n=\theta}^{t_i} [g(t_n, \theta)^T \Lambda(t_n)^{-1} \gamma(t_n)] \quad (2.58)$$

$$S(t_i, \theta) = \sum_{t_n=\theta}^{t_i} [g(t_n, \theta)^T \Lambda(t_n)^{-1} g(t_n, \theta)] \quad (2.59)$$

Now it is necessary to remove the dependence on ν from Equation (2.57). This is done by finding the Maximum Likelihood Estimate(MLE) of ν for Equation (2.57). This is accomplished by taking the derivative of Equation (2.57) with respect to the variable ν and setting it equal zero:

$$\begin{aligned}\frac{d[\ln(L(t_i, \theta, \nu))]}{d\nu} &= \Psi(t_i, \theta) - \nu S(t_i, \theta) = 0 \\ \hat{\nu} &= \frac{\Psi(t_i, \theta)}{S(t_i, \theta)}\end{aligned}\tag{2.60}$$

Now that the MLE of ν has been found over the interval θ , substitute this value back into Equation (2.57). This yields the following maximum likelihood equation:

$$\ln(L(t_i, \theta, \nu)) = l(t_i, \theta) = \frac{\Psi^2(t_i, \theta)}{2S(t_i, \theta)}\tag{2.61}$$

Now the expression for $l(t_i, \theta)$ is maximized over θ to obtain the *maximum* likelihood estimate. This is the final form of the Generalized Likelihood Ratio test. If $l(t_i, \theta)$ exceeds a predetermined threshold T then a failure of type $n(t_i, \theta)$ will be declared on the measurement specified in $d(t_i)$. The GLR magnitude estimate of this failure will be $\hat{\nu}$ as defined in Equation (2.60).

2.3.2.2 Development of the GLR Matching Filter. While the GLR algorithm defined in the previous section is designed to function as a stand-alone failure detection and isolation algorithm, the GLR matching filter algorithm performs a more limited role within the MNRS model. The GLR matching filters will be provided with failure detection and isolation information from the bank of Chi-Square tests that are run on each NRS filter. Given the failure information from the Chi-Square tests, the GLR matching filter must then identify the type of failure. Specifically, the GLR matching filter algorithm will attempt to differentiate between step and ramp failure offsets. There is no GLR failure model available for the noise offset, therefore a Generalized Likelihood Ratio is not developed for the noise offset. Fundamentally this is done by comparing the results of different GLR matching filters, and choosing which best fits the residuals. Each GLR matching filter can also estimate the magnitude of the failure. This section derives the

GLR matching filters from the previously defined GLR equations, and explains how the matching filters accomplish failure identification and estimation. Before this can be done however, the information provided by the Chi-Square test must be defined.

The first piece of information is the identity of the failed measurement signal. The knowledge of which measurement update is failed, allows the GLR vector $d(t_i)$ to become the fixed vector $d_\chi(t_i)$ (subscript χ signifies dependence on Chi-Square failure information). The second piece of information provided by the Chi-Square test is its best estimate of when the failure occurred. This estimate is based on experimental analysis of Chi-Square detection delay. The knowledge of the Chi-Square estimate allows the variable θ to become the known estimate $\hat{\theta}_\chi$.

These two simplifications greatly reduce the computational burden of the GLR algorithm. In fact the extra information reduces the GLR equations into simple matching filter equations. Rather than having a maximum likelihood estimate over the variable θ , the GLR equation reduces to the following;

$$l_{mod}(t_i) = \frac{\Psi_\chi^2(t_i, \hat{\theta}_\chi)}{2S_\chi(t_i, \hat{\theta}_\chi)} \quad (2.62)$$

with $d_\chi(t_i)$ and $\hat{\theta}_\chi$ predetermined by the MNRS Chi-Square tests. The GLR equation has now been reduced to a function of only time, t_i .

To implement the failure type identification, two GLR matching filters are constructed; one for a step bias and one for a ramp offset. These filters are unique because they are each driven by two different failure models, $n_{step}(t_i, \hat{\theta}_\chi)$ and $n_{ramp}(t_i, \hat{\theta}_\chi)$. The definitions for the different failure models are provided in Section 3.4.3. It is important to note that the Chi-Square estimate of the failure time, $\hat{\theta}_\chi$, does not take into account the detection delay inherent in the step and especially the ramp offsets. Compensating for this failure detection delay in the failure models is addressed in Section 3.4.5. Continuing on, for each of the two failure models, n_{step} and n_{ramp} , a GLR matching filter is now defined as,

$$l_{step}(t_i) = \frac{\Psi_{step}^2(t_i, \hat{\theta}_\chi)}{2S_{step}(t_i, \hat{\theta}_\chi)} \quad (2.63)$$

$$l_{ramp}(t_i) = \frac{\Psi_{ramp}^2(t_i, \hat{\theta}_\chi)}{2S_{ramp}(t_i, \hat{\theta}_\chi)} \quad (2.64)$$

where the first equation is a function of $n_{step}(t_i, \hat{\theta}_\chi)$ and the second equation is a function of $n_{ramp}(t_i, \hat{\theta}_\chi)$. A ratio of these two equations is defined as $\Upsilon(t_i)$,

$$\Upsilon(t_i) = \frac{l_{ramp}(t_i, \hat{\theta}_\chi)}{l_{step}(t_i, \hat{\theta}_\chi)} \quad (2.65)$$

This equation simplifies through cancellation and use of Equations (2.58),(2.59),and (2.61) to reach the following result,

$$\Upsilon(t_i) = \frac{\Psi_{ramp}^2(t_i, \hat{\theta}_\chi)S_{step}(t_i, \hat{\theta}_\chi)}{\Psi_{step}^2(t_i, \hat{\theta}_\chi)S_{ramp}(t_i, \hat{\theta}_\chi)} \quad (2.66)$$

$$\Upsilon(t_i) = \frac{(\sum_{t_n=\hat{\theta}_\chi}^{t_i} [g_{ramp}^T(t_n, \hat{\theta}_\chi)\Lambda(t_n)^{-1}\gamma(t_n)])^2 \sum_{t_n=\hat{\theta}_\chi}^{t_i} [g_{step}(t_n, \hat{\theta}_\chi)^T \Lambda(t_n)^{-1} g_{step}(t_n, \hat{\theta}_\chi)]}{(\sum_{t_n=\hat{\theta}_\chi}^{t_i} [g_{step}(t_n, \hat{\theta}_\chi)^T \Lambda(t_n)^{-1}\gamma(t_n)])^2 \sum_{t_n=\hat{\theta}_\chi}^{t_i} [g_{ramp}(t_n, \hat{\theta}_\chi)^T \Lambda(t_n)^{-1} g_{ramp}(t_n, \hat{\theta}_\chi)]} \quad (2.67)$$

where the subscripts, ramp and step, correspond to a function that has the respective failure model as an input. Similar to the original GLR algorithm, the ratio of matching filters, $\Upsilon(t_i)$, will be compared to thresholds to determine which type of failure has occurred. If the failure is a ramp bias, the ratio of matching filters should be much greater than the one, therefore $\Upsilon(t_i)$ should be much larger than one. Likewise $\Upsilon(t_i)$ should be much less than one if a step bias has occurred. Yet if the failure is additive random noise, neither of the failure models should match the failure, and $\Upsilon(t_i)$ should remain approximately one. Therefore two failure type thresholds will be established for $\Upsilon(t_i)$,

$$\text{Failure Type} = \begin{cases} \text{Ramp Bias} & \text{if } \Upsilon(t_i) > T_1 \\ \text{Step Bias} & \text{if } \Upsilon(t_i) < T_2 \\ \text{Noise Bias} & \text{otherwise} \end{cases} \quad (2.68)$$

The $T_1 \gg 1$, ramp bias, and $T_2 \ll 1$, step bias thresholds are experimentally established thresholds. With the proper choice of thresholds the $\Upsilon(t_i)$ will accurately determine failure type.

The magnitude of the failure type is determined from Equation (2.60). This equation, calculated for both step and ramp failure models, will estimate the size of the failure at each point in time, $t_i > \theta_x$. The determination of which estimate is valid depends on the outcome of the ratio between matching filters, $\Upsilon(t_i)$. By using Equations (2.60) and (2.66), accurate estimates can be made of both the size of a step bias, and the slope of a ramp offset. Unfortunately, since there is no failure model for the noise bias, an estimate of the magnitude of a noise bias cannot be determined.

This section has defined the role of the GLR algorithm for the MNRS FDIR system. The GLR has been modified to use predetermined information from the MNRS Chi-Square test to ease computational burden and increase the likelihood of accurate estimation. While theoretically the GLR matching filters should effectively accomplish failure identification and estimation, past AFIT research has only had limited success with the GLR algorithm. Therefore, a second matching filter algorithm is also developed to compare with the GLR matching filters.

2.3.3 The Chi-Square Pattern Recognition(CSPR) Matching Filter. The CSPR filter identifies consistent trends in Chi-Square magnitude plots that can distinguish failure type. The filter is based on the premise that step and ramp offsets affect the residuals in distinct manners that are obvious in the Chi-Square results. By examining the Chi-Square magnitude data over the time of failure, the type of failure is determined. Like the GLR algorithm, it is assumed that the Chi-Square estimate of the failure and the identity of the failed measurement are known. To understand the implementation of the CSPR matching filter algorithm, the scope of the filter's theoretical development needs to be clarified.

The CSPR algorithm has been developed through post-process examination of the impact of failures on the residual and Chi-Square plots. These Chi-Square results and the results of past FDIR EKF work imply that the failure type can be determined by the shape of the Chi-Square test over the time of failure [2, 5]. This thesis in no way implies that the CSPR algorithm can be used to identify failures for all EKF models. The CSPR filter has been designed for the specific NRS model developed at AFIT. Fundamentally, the CSPR filter is an ad-hoc identification algorithm that exploits the engineering insight developed

through rigorous failure testing of a filter model. The criteria for distinguishing the failure type in the residuals is presented in Section 3.4.4.

2.3.4 Threshold Selection and Filter Tuning. Threshold choosing is a trade-off between several performance parameters to gain the best FDI for the system. The performance parameters that must be balanced are false alarm rate, detection delay, and missed failures. If the threshold value is high, the false alarm rate will be low, while the detection delay and the missed failure rate will be high. If the threshold value is lower, the false alarm rate will increase, while the detection delay and missed failure rate will decrease. The system dynamics must also be taken into account when choosing a threshold. A highly dynamic system will require a higher threshold than a benign system to maintain an acceptable false alarm rate. In the end, a balance is achieved between the performance parameters that yields the best possible FDI for the problem at hand. The threshold choices made for the MNRS are discussed in Chapters III and IV.

The tuning of the Kalman filter can also be varied to increase FDI ability. By altering the tuning of the Kalman filter, the residuals can become more sensitive to failures. The effect of changing the tuning can be seen in Equation (2.29). If the value of R were to decrease, the magnitude of the Chi-Square test would actually be increased since the inverse of the covariance is used in the Chi-Square computation, Equation (2.30). Therefore, it is possible to make a system more sensitive to failures by adjusting the tuning. Yet by altering the tuning, the filter will no longer be constructed to yield the optimal navigation solution, i.e., best state estimates. Therefore varying the tuning parameters is not implemented to improve the ability of the FDI algorithms. This research maintains the most accurate navigation solution from the MNRS model.

2.4 Chapter Summary

Chapter II presented the Kalman filter, the Chi-Square test, and the GLR test in support of the MNRS failure detection and isolation algorithm. The first theory presented, the Kalman filter theory, will be more deeply explored in the beginning of Chapter III with the presentation of the navigation models. The use of the two failure algorithms will also

be explored in Chapter III with a description of the implementation of the MNRS FDI algorithm.

III. Navigation and Failure Models

This chapter describes the models used for the computer simulation of the MNRS FDIR algorithm. The chapter begins with a high-level description of the MNRS and its major sub-components. Following this introduction, the NRS filter is described. The description of the NRS filter focuses on the implementation of the navigation models in the MSOFE software [13]. After the NRS filter description, the implementation of the failure models is presented. The chapter concludes with a discussion of the MNRS filter selection algorithm.

3.1 The MNRS Model Description

Figure 3.1 reviews the major sub-components within the MNRS. To the far left in Figure 3.1, the GPS and RRS measurement sources can be found. These two navigation aids distribute unique sets of measurement information to each of the ten NRS extended Kalman filters. Table 1.1 provides an exact list of the measurement signals each NRS filter receives. The navigation solution comes from the INS. Using the information from the GPS, RRS, and INS, each NRS filter calculates its best estimate of the navigation errors made by the INS. The ten chi-square tests monitor the residuals of all the NRS filters and pass on the best navigation solution from the bank of NRS filters. When the chi-square test declares that a measurement signal has been corrupted, the navigation correction switches to the unaffected filter. The Chi-Square test also passes on the identity and the estimate of failure time to the matching filter. The matching filter then estimates the type of failure. This brief overview shows how the INS, the GPS receiver, the RRS receiver, the NRS EKFs, the chi-square tests, and the matching filter all work together to produce a navigation solution that is robust against failures in the GPS and RRS measurement signals.

While the major focus of this thesis is the FDIR capability of the MNRS, the quality of the *navigation* system fundamentally rests on the ability of each NRS filter to provide an accurate estimate of the errors committed by the INS in calculating its navigation solution.

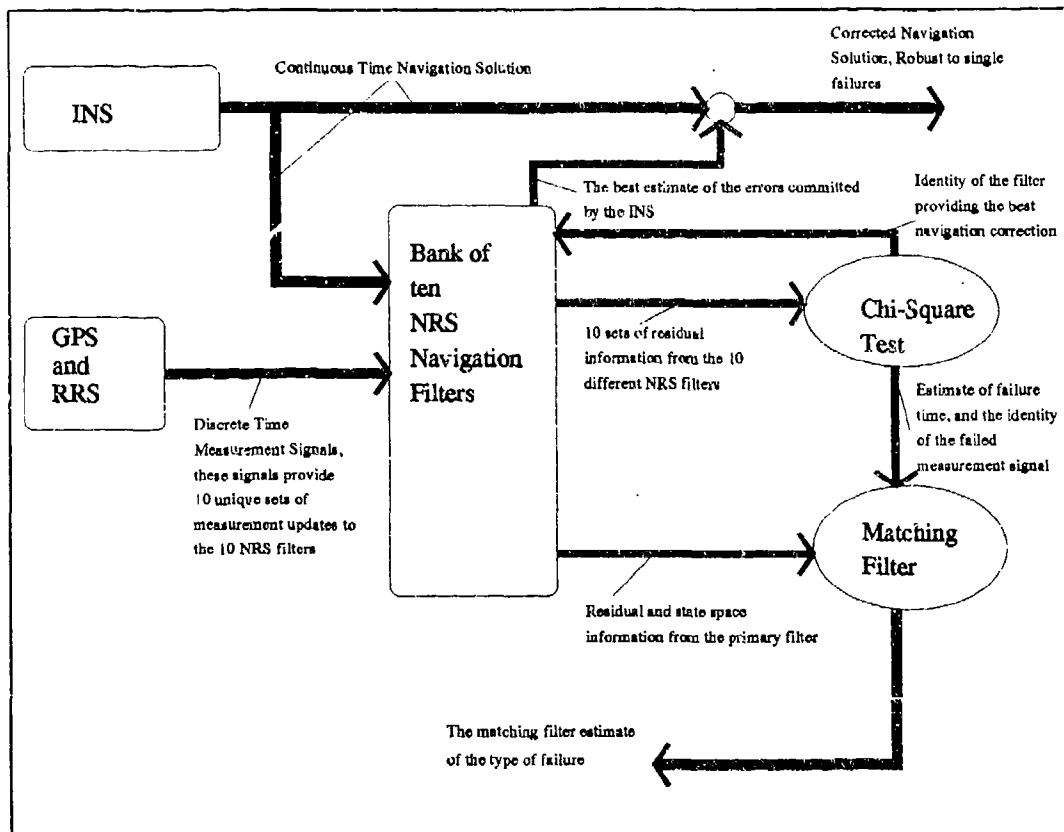


Figure 3.1 Overall MNRS Description

The next section will fully develop the NRS extended Kalman filter equations and their implementation in the MSOFE software.

3.2 The NRS Computer Model

The information and data presented in this thesis has been accomplished entirely through computer simulation. The computer modelling of the NRS system is divided into two portions, the *truth model* and the *filter model*. The *truth model* is a computer-generated real world for the NRS filter. The truth model generates such things as the measurement updates for the NRS filter, the true flight profile of the aircraft, and a state variable baseline for evaluating filter performance. The truth model consists of 95 error states about their nominal values. The filter model represents the NRS model in its functional form. The

filter model is a 15-state extended Kalman filter that has been developed through order reduction of the 95-state truth model.

The block diagram, Figure 3.2, explains how the filter and truth models interact in the MSOFE computer simulation. PROFGEN provides a simulated flight profile and ORBIT generates GPS satellite constellation positions. With this information, the truth model is able to simulate the *real world* INS navigation solution $x + \delta x_{INS}$ and generate the *real world* GPS and RRS measurements, R_{GPS} and R_{RRS} respectively. The NRS filter in Figure 3.2 is represented by the extended Kalman filter block. The corrections from the NRS filter are subtracted from the INS navigation solution to generate the best possible navigation solution available, $\hat{x} = x + \delta x - \delta \hat{x}$. Now that the MSOFE implementation of the NRS filter has been explained, the truth and filter models for the GPS, RRS, and INS subsystems will be described.

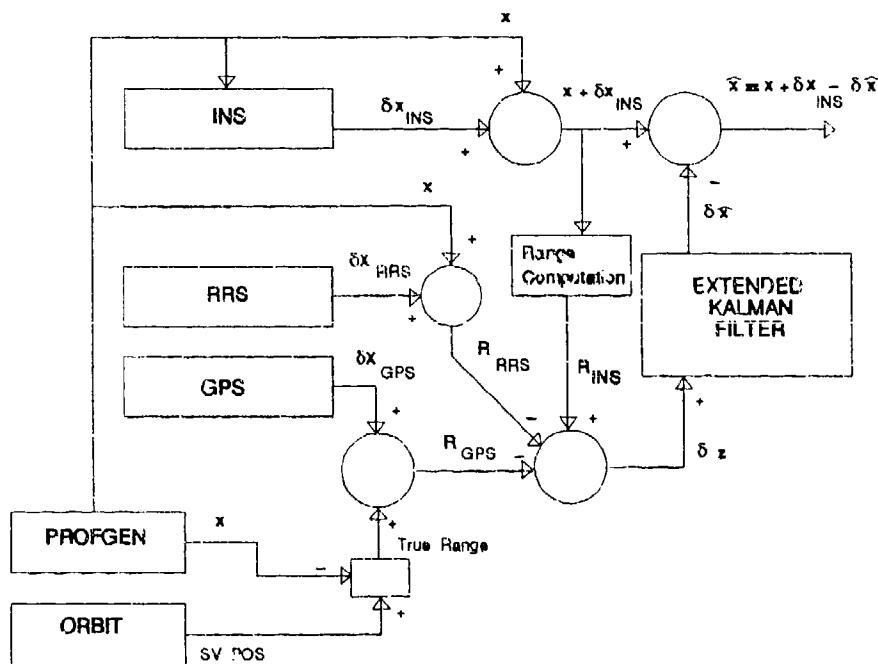


Figure 3.2 Truth and Filter Model Block Diagram

3.3 NRS Model Description

This section presents the truth and the filter model EKF propagation and measurement equations (Equations (2.9) and (2.11), respectively). The presentation will be divided up by navigation subsystems. First the INS portion of the equations will be presented, then the RRS, followed by the GPS. Furthermore, before the different navigation subsystems are individually described, the high-level state and measurement equations are provided for the NRS filter followed by the truth model.

Equations (3.1) and (3.2) show how the different navigation subsystem models combine to form a single NRS filter model:

$$\delta \dot{x}_f = \begin{bmatrix} F_{INS_f} & 0 & 0 \\ 0 & F_{RRS_f} & 0 \\ 0 & 0 & F_{GPS_f} \end{bmatrix} \delta x_f + \begin{bmatrix} w_{INS_f} \\ w_{RRS_f} \\ w_{GPS_f} \end{bmatrix} \quad (3.1)$$

$$\delta z_f = \begin{bmatrix} H_{INS_f} \\ H_{RRS_f} \\ H_{GPS_f} \end{bmatrix} \delta x_f + \begin{bmatrix} v_{INS_f} \\ v_{RRS_f} \\ v_{GPS_f} \end{bmatrix} \quad (3.2)$$

As stated earlier, the overall filter model consists of 15 states; 11 INS, 2 RRS, and 2 GPS states. Table A.5 in Appendix A provides a description of the 15-state vector, δx_f , implemented in the filter model. References to further descriptions of the sub-matrices in the filter equations can be found in Table 3.1. It should be noted that the barometric altimeter and velocity aiding measurements are considered to be INS measurements, while the GPS and RRS range measurements are the respective updates for the GPS and RRS.

The propagation and measurement equations for the NRS *truth* model is presented in similar fashion below:

$$\delta \dot{x}_t = \begin{bmatrix} F_{Filter} & F_{INS_{t1}} & 0 & 0 \\ 0 & F_{INS_{t2}} & 0 & 0 \\ 0 & 0 & F_{RRS_t} & 0 \\ 0 & 0 & 0 & F_{GPS_t} \end{bmatrix} \delta x_t + \begin{bmatrix} w_{Filter} \\ w_{INS_t} \\ w_{RRS_t} \\ w_{GPS_t} \end{bmatrix} \quad (3.3)$$

Table 3.1 References for the Sub-Matrices of the NRS Truth and Filter Models

Filter Model	Location of Description	Truth Model	Location of Description
F_{INS_f}	Section 3.3.1.3	$F_{INS_{filter}}$	Section 3.3.1.3,3.3.2.1,3.3.3.1
-	-	$F_{INS_{t1}}$	Section 3.3.1.2
-	-	$F_{INS_{t2}}$	Section 3.3.1.2
F_{RRS_f}	Section 3.3.2.2	F_{RRS_t}	Section 3.3.2.1
F_{GPS_f}	Section 3.3.3.2	F_{GPS_t}	Section 3.3.3.1
w_{INS_f}	Section 3.3.1.3	w_{INS_t}	Section 3.3.1.2
w_{RRS_f}	Section 3.3.2.2	w_{RRS_t}	Section 3.3.2.1
w_{GPS_f}	Section 3.3.3.2	w_{GPS_t}	Section 3.3.3.1
H_{INS_f}	Section 3.3.1.4	H_{INS_t}	Section 3.3.1.4
H_{RRS_f}	Section 3.3.2.3	H_{RRS_t}	Section 3.3.2.3
H_{GPS_f}	Section 3.3.3.3	H_{GPS_t}	Section 3.3.3.3

$$\delta z_t = \begin{bmatrix} H_{INS_t} \\ H_{RRS_t} \\ H_{GPS_t} \end{bmatrix} \delta x_t + \begin{bmatrix} v_{INS_t} \\ v_{RRS_t} \\ v_{GPS_t} \end{bmatrix} \quad (3.4)$$

The NRS truth model consists of the original fifteen states of the filter model (represented by F_{Filter} and w_{Filter}), augmented by additional INS, GPS, and RRS states. The total number of states for the three navigation subsystems is 95; 39 INS states, 26 RRS states, and 30 GPS states. Tables A.1-A.4, in Appendix A give a full description of each individual state of the truth model. Also Tables B.1-B.5 and Tables B.6-B.7 in Appendix B have a complete listing of the components of the F and the Q noise strengths associated with the W matrices in Equation (3.3).

While the first fifteen truth states of the filter model are nearly identical to the first fifteen states of the model, there is one crucial difference. The filter model dynamics driving noise and measurement noise do *not* correlate with those of the first fifteen states of the truth model. The filter model noise values have been altered to achieve good tuning against the truth model [9]. The following sections will provide a detailed presentation into the exact make-up of the truth and filter model propagation and measurement equations for all three navigation subsystems.

3.3.1 The Inertial Navigation System(INS) Model. This section presents the truth and filter models used for the INS. The INS model is a strapped-down wander

azimuth system that senses aircraft motion via gyros and accelerometers and is used as the primary source for navigation. The INS model has been derived from the Litton 93-State model [4]. First, the original 93-state Litton model will be presented, followed by the reduced-ordered 39-state truth and 11-state filter models. After the truth and filter state equations have been defined, barometric altimeter and Doppler velocity aiding INS measurement equations will be presented.

3.3.1.1 The 93-State LN-93 Error Model. The 93-state Litton INS MSOFE computer model has been generated by the Wright Labs Avionics Directorate Research Group. Their development uses both past AFIT research and Litton documentation to fine tune past modelling efforts [4, 14, 18, 19]. The 93-state model generates a high number of documented error sources that are found in the Litton wander-azimuth LN-93 INS. These errors are described using six categories of states:

$$\delta x = [\delta x_1^T \quad \delta x_2^T \quad \delta x_3^T \quad \delta x_4^T \quad \delta x_5^T \quad \delta x_6^T]^T \quad (3.5)$$

where δx is a 93×1 column vector and:

δx_1 represents the "general" error vector containing 13 position, velocity, attitude, and vertical channel errors.

δx_2 consists of 16 gyro, accelerometer, and baro-altimeter exponentially time-correlated errors, and "trend" states. These states are modeled as first order Markov processes in the truth (system) model.

δx_3 represents gyro bias errors. These 18 states are modeled as random constants in the truth model.

δx_4 is composed of the accelerometer bias error states. These 22 states are modeled in exactly the same manner as the gyro bias states.

δx_5 depicts accelerometer and gyro initial thermal transients. The 6 thermal transient states are first order Markov processes in the system model.

δx_6 models the gyro compliance errors. These 18 error states are modeled as biases in the system model.

The 93-State Litton model state space differential equation is given by:

$$\begin{Bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \\ \delta \dot{x}_3 \\ \delta \dot{x}_4 \\ \delta \dot{x}_5 \\ \delta \dot{x}_6 \end{Bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\ 0 & F_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & F_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \\ \delta x_5 \\ \delta x_6 \end{Bmatrix} + \begin{Bmatrix} w_1 \\ w_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (3.6)$$

A full description of the sub-matrices for this equation is given in the Litton LN-93 manual [4]. This large state model represents the most accurate model available for the LN-93 navigation errors.

3.3.1.2 The 39-State INS Truth Model. While the 93-state model is a very accurate representation of the INS error characteristics, the high dimensionality of the state equations makes the model impractical for failure detection analysis. Previous AFIT theses have demonstrated that reduced-ordered truth models can be used in place of the 93-State model without losing a significant degree of accuracy [6, 14]. Therefore the INS truth model has been reduced to a 39-state model. The reduced-ordered model retains only the truly essential states from Equation (3.6). The truth model state space equation is defined in Equation (3.7):

$$\begin{Bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \\ \delta \dot{x}_3 \\ \delta \dot{x}_4 \end{Bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ 0 & F_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{Bmatrix} + \begin{Bmatrix} w_1 \\ w_2 \\ 0 \\ 0 \end{Bmatrix} \quad (3.7)$$

It should be noted that the INS truth state vector δx , is a 39-state vector. The four components of δx do not directly correlate to the first four components of the 93-state

Litton model. For a complete listing of the 39 states and how they relate to those in the Litton model, see Tables A.1 and A.2 in Appendix A.

3.3.1.3 The 11-State INS Filter Model. The INS filter model retains the essential states from the 39-state truth model. Through past AFIT research, the 11-state INS filter has been shown to perform adequately when given frequent GPS and RRS measurement updates [14, 19]. Table A.5 in Appendix A shows the 11 states used for the INS filter model. Dynamics driving noise has also been added to every state to compensate for the order-reduction of the model. The final INS filter state dynamic driving noise values can be found in Table B.8 of Appendix B.

3.3.1.4 INS Measurement Models. The two INS measurements that are used to update the filter are the barometric altimeter and the Doppler based velocity. Both these signals are used to correct for inherent instabilities in the filter. First the altimeter measurement will be presented, followed by the Doppler based velocity measurement. It should be noted that since the NRS filter is an error state filter, it is necessary to develop *difference* measurement update equations for all the measurements. The altimeter measurement equation is based on the difference between the INS-predicted altitude, Alt_{INS} and the barometric altimeter-predicted altitude Alt_{bar} :

$$\delta z_{Alt} = Alt_{INS} - Alt_{bar} \quad (3.8)$$

Therefore it is necessary to develop the two separate measurement signals that will be *differenced* to attain the proper measurement update for the error state filter. The INS-predicted altitude is the sum of the true altitude, h_t , and the INS error in vehicle altitude above the reference ellipsoid, δh . The barometric altimeter reading is modelled as the sum of the true altitude, h_t , the total error in the barometric altimeter, δh_B , and a random measurement noise, v . The difference measurement update signal is formed in Equation (3.9) by subtracting the INS-predicted altitude from the barometric altimeter altitude:

$$\delta z_{Alt} = Alt_{INS} - Alt_{Bar}$$

$$\begin{aligned}
&= [h_t + \delta h] - [h_t + \delta h_B + v] \\
&= \delta h - \delta h_B + v
\end{aligned} \tag{3.9}$$

A perfect Doppler system provides velocity aiding to the INS based on assumption 12 in Chapter I. The Doppler measurement has been added to lend numerical stability to the NRS filter. A simple model is assumed for the Doppler measurement. All three channels (north, east, and up) are represented by the difference between the *truth* state velocity error, δV_{t_i} , and the *filter* state velocity error, δV_{f_i} , as shown in Equation (3.10).

$$\delta z_i = \delta V_{t_i} - \delta V_{f_i} \quad \text{where } i = x, y, z \tag{3.10}$$

Although this model seems somewhat unrealistic, in that it provides the filter with an ideal difference measurement for velocities, it does not skew the performance of the FDI algorithm because these measurements are not used in the FDI calculations. The simplistic model is being used until a more accurate model can be developed. Use of this model also allows the results of this thesis to be directly compared to past AFIT research [14, 17, 18, 19].

There is no difference in the form of the INS truth and the filter model measurement equations. Both use the same states to create the INS difference measurements. The only distinction lies in the measurement noise values for each of the two equations. The truth and filter measurement noise values are located in Table B.9 of Appendix B. This completes the presentation of the INS truth and filter state equations as well as the INS measurement equations. The next section will develop similar equations for the ground-based transponder system used in this thesis.

3.3.2 The Range/Range-Rate System (RRS) Model. The RRS system is the primary CIGTF ground-based transponder system that has been installed for testing and evaluating navigation equipment [14, 18]. The RRS *interrogates* the transponders, collecting the electro-magnetic (EM) signals they emit. These signals give the user the range to the transponder. In this thesis, the navigation information passed to the NRS filters is the

range to five transponders and the known location of those transponders. As done with the INS models, first the RRS truth model state equations will be presented, followed by the filter model state equations, and finally the measurement update equations.

3.3.2.1 26-State RRS Truth Model. The RRS truth model contains 26 states to simulate the real world errors that exist in the transponder system. The first two error states are *common* error states, i.e., these errors are common to all of the transponders. The two common states for the transponders are a result of errors in user hardware. They appear as bias terms and are modeled as random constants. These state equations are given by:

$$\begin{Bmatrix} \dot{\delta R_b} \\ \dot{\delta v_b} \end{Bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_b \\ \delta v_b \end{Bmatrix} \quad (3.11)$$

where

$$\begin{aligned} \delta R_b &= \text{Range error due to equipment bias} \\ \delta v_b &= \text{Velocity error due to equipment bias} \end{aligned}$$

The initial conditions for the truth model states were chosen to be consistent with CIGTF and previous AFIT research [14, 17, 18, 19] and are:

$$\begin{Bmatrix} \delta \hat{R}_b(t_0) \\ \delta \hat{v}_b(t_0) \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.12)$$

and

$$\mathbf{P}_{\delta R_b, \delta v_b}(t_0) = \begin{bmatrix} 1 f^2 & 0 \\ 0 & 10^{-4} f t^2 / sec^2 \end{bmatrix} \quad (3.13)$$

Along with the two common error states for the transponder, each of the transponder signals has four error states to model the unique errors of that specific transponder unit. These errors represent the error in the surveyed position (x, y, and z) of the transponder's location and the atmospheric propagation delay between the transponder and the receiving aircraft. The position errors are modeled as random constants and the atmospheric error is represented by a first order Markov process. The state equations for these error sources

are shown below where i represents the various transponders (1-6):

$$\begin{Bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ \dot{\delta R_{atm,i}} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{300sec} \end{bmatrix} \begin{Bmatrix} x_i \\ y_i \\ z_i \\ \delta R_{atm,i} \end{Bmatrix} + \begin{Bmatrix} w_{x,i} \\ w_{y,i} \\ w_{z,i} \\ w_{atm,i} \end{Bmatrix} \quad (3.14)$$

The initial conditions for these states were chosen to be consistent with previous AFIT research [14, 17, 18]. The initial mean and variance for the truth model are defined below.

$$\hat{x}_{x,y,z,atm}(t_0) = 0 \quad (3.15)$$

$$P_{x,y,z,atm}(t_0) = \begin{bmatrix} 25ft^2 & 0 & 0 & 0 \\ 0 & 25ft^2 & 0 & 0 \\ 0 & 0 & 25ft^2 & 0 \\ 0 & 0 & 0 & 100ft^2 \end{bmatrix} \quad (3.16)$$

The truth model dynamics driving noise that has been implemented for the transponder error states was provided by Holloman test facilities through past AFIT research [14, 17, 18, 19]. The mean and variance of the dynamic driving noise are:

$$E\{w_{x,y,z,atm}(t)\} = 0 \quad (3.17)$$

$$E\{w_{x,y,z,atm}(t)w_{x,y,z,atm}^T(t+\tau)\} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2\sigma_{atm}^2}{300} \end{bmatrix} \delta(\tau) \quad (3.18)$$

respectively, with $\sigma_{atm}^2 = 10^{-10} \frac{ft^2}{sec}$. Therefore the truth model will consist of 26 states, two common user error states plus six sets of four unique transponder error states. It should be noted that the truth model as simulated will propagate six sets of unique transponder error states, yet only five are used by each NRS filter. The extra state has been maintained to ease the implementation of the MNRS model. However it is important to understand

that each individual NRS filter receives only five RRS transponder signals. This completes the description of the RRS truth model. Next the RRS filter model will be presented.

3.3.2.2 The 2-State RRS Filter Model. Research at AFIT has shown that retaining only the first two states, which are common to all the transponders, provides good filter performance [19]. These states are the range and velocity errors due to equipment bias and are represented as δR_b and δv_b in Equation (3.11). To compensate for the removal of the other states and to prevent EKF gains from going to zero, dynamics driving noise has been added to each of the two RRS filter states.

$$\begin{Bmatrix} \delta \dot{R}_b \\ \delta \dot{v}_b \end{Bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_b \\ \delta v_b \end{Bmatrix} + \begin{Bmatrix} w_{\delta R_b} \\ w_{\delta v_b} \end{Bmatrix} \quad (3.19)$$

The initial mean and variance for the filter model is assumed to be zero for these two states. Filter tuning accounted for the exact magnitudes used for the strengths of the dynamics driving noise, $w_{\delta R_b}$ and $w_{\delta v_b}$. The final values implemented can be found in Table B.8 of Appendix B.

3.3.2.3 RRS Measurement Model. Each NRS filter will receive five different measurement update signals from the five different transponder locations. This section describes the measurement equations for the RRS transponder updates. Each equation for the transponder updates is identical in form. First the truth model measurement equation will be fully developed, followed by a brief description of the reduced-order filter model measurement equation.

The RRS difference measurement is generated by forming two independent measurements of the range from the transponder to the aircraft. The EKF then takes the difference of these two measurements to form δz . The two range measurements differenced are the INS-computed range (R_{INS}) and the RRS-calculated range, (R_{RRS}):

$$\delta z_{RRS} = R_{INS} - R_{RRS} \quad (3.20)$$

The RRS range measurement, R_{RRS} is the sum of the true range from the transponder to the aircraft and the errors inherent in the measurement signal:

$$R_{RRS} = R_t + \delta R_{atm} + \delta R_b + v \quad (3.21)$$

where

R_{RRS}	=	RRS range measurement, from transponder to user
R_t	=	true range, from transponder to user
δR_{atm}	=	range error due to atmospheric delay
δR_b	=	error due to equipment bias
v	=	zero-mean white Gaussian measurement noise

The second source of range information is provided by the INS. The NRS takes the INS computed position, X_U and subtracts the known transponder position, X_T . Equation (3.22) shows the calculation of the R_{INS} :

$$R_{INS} = \left| X_U - X_T \right| = \left| \begin{Bmatrix} x_U \\ y_U \\ z_U \end{Bmatrix} - \begin{Bmatrix} x_T \\ y_T \\ z_T \end{Bmatrix} \right| \quad (3.22)$$

It should be noted that both X_T and X_U represent position vectors in the earth-centered earth-fixed (ECEF) frame. While Equation (3.22) provides the second source of transponder range information, this equation is not a function of the EKF error state variables. Therefore it is necessary to rewrite Equation (3.22) in the following form.

$$R_{INS} = \sqrt{(x_U - x_T)^2 + (y_U - y_T)^2 + (z_U - z_T)^2} \quad (3.23)$$

Equation (3.23) is equivalent to Equation (3.22). This new equation will now be approximated to generate a relationship for R_{INS} that is a function of the EKF error state variables. Based on assumption 11 from Chapter I, Equation (3.23) can be approximated and rewritten in terms of the true range and a truncated first-order Taylor series, with

perturbations representing the errors in X_U and X_T :

$$R_{INS} = R_t + \left. \frac{\partial R_{INS}(X_T, X_U)}{\partial X_U} \right|_{(X_T, X_U)_{nom}} \cdot \delta X_U + \left. \frac{\partial R_{INS}(X_T, X_U)}{\partial X_T} \right|_{(X_T, X_U)_{nom}} \cdot \delta X_T \quad (3.24)$$

The final expression for R_{INS} , Equation (3.25), is found by evaluating the partial derivatives in Equation (3.24). It should be noted that Equation (3.25) is a function of δx_U , δy_U , δz_U , δx_T , δy_T , and δz_T . These variables can be directly transformed into the state variables in the EKF truth state equation.

$$R_{INS} = R_t - \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_U - \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_U - \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_U + \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_T + \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_T + \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_T \quad (3.25)$$

The actual *difference* equation, Equation (3.26) is now formed by taking the difference between R_{INS} and R_{RRS}

$$\begin{aligned} \delta z_{RRS_i} &= R_{INS} - R_{RRS} \\ &= - \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_U - \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_U - \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_U \\ &\quad + \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_T + \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_T + \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_T \\ &\quad - [1] \delta R_{atm} - [1] \delta R_b - v \end{aligned} \quad (3.26)$$

The equation for the difference measurement, Equation (3.26), is the complete form of the measurement for the truth model. In the above equation, δx_U , δy_U , and δz_U can be translated through an orthogonal transformation into the state variables $\delta \theta_x$, $\delta \theta_y$, and $\delta \theta_z$ defined as truth model states in Table A.1 of Appendix A [1].

The actual measurement equation that is used for the reduced-order NRS filter does not retain the terms δx_T , δy_T , and δz_T , since the reduced-ordered model does not contain these states. Therefore with those values set to zero Equation (3.27) is the filter model

measurement equation for the NRS filter.

$$\begin{aligned}
 \delta z_{RRS_j} &= R_{INS} - R_{RRS} \\
 &= - \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_U - \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_U - \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_U \\
 &\quad - [1] \delta R_b - v
 \end{aligned} \tag{3.27}$$

The filter measurement noise strength, R will be tuned to attain adequate performance despite the reduction in order from the truth model and the Taylor series approximation. Table B.9 in Appendix B contains a complete listing of the measurement noise values for both the filter and the truth model models. This ends the description of the RRS navigation subsystem. Next the GPS navigation equations are presented.

3.3.3 The Global Positioning System(GPS) Model. The third and final navigation system is based on EM signals transmitted from orbiting GPS satellites. Although similar in concept to the RRS, the GPS is modeled somewhat differently. This model has been developed through research at AFIT, and many of its fundamental concepts are addressed in a variety of sources [7, 14, 17, 18]. GPS is similar to the RRS in that it generates navigation information by acquiring the range to multiple satellites of known position. The navigation information passed to each NRS filters is the range to four satellites and the ephemeris data position of those four satellites [7]. The next three sections will present all the necessary equations to define the GPS truth and filter models fully.

3.3.3.1 The 30-State GPS Truth Model. There are five types of error sources that are modeled in the GPS truth model state equations. The first two states represent the errors in the user clock and are modeled as follows:

$$\begin{Bmatrix} \delta \dot{R}_{clk_U} \\ \delta \dot{D}_{clk_U} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{clk_U} \\ \delta D_{clk_U} \end{Bmatrix} \tag{3.28}$$

where

$$\delta R_{clk_U} = \text{range equivalent of user clock bias}$$

δD_{clk_U} == velocity equivalent of user clock drift

The initial state estimates and covariances for these states were chosen to be consistent with previous AFIT research [14, 17, 18] and are:

$$\begin{Bmatrix} \hat{\delta R}_{clk_U}(t_0) \\ \hat{\delta D}_{clk_U}(t_0) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (3.29)$$

and

$$P_{\delta R_{clk_U}, \delta D_{clk_U}}(t_0) = \begin{bmatrix} 9.0 \times 10^{14} ft^2 & 0 \\ 0 & 9.0 \times 10^{10} ft^2/sec^2 \end{bmatrix} \quad (3.30)$$

Because these error sources are a function of the user equipment, they are common to all the satellite vehicles. The remaining five sources of errors are unique to each satellite vehicle (SV), based on their individual equipment and their position with respect to the user. The first SV-specific error source is the code loop error. Although the code loop is part of the user equipment shared by all the SV's, its error magnitude is relative to each SV. The second and third SV-specific errors are the atmospheric interference with the EM signals, as related to the ionospheric and tropospheric delay in the signal's propagation. The code loop error, tropospheric delay, and ionospheric delay are all modeled as first order Markov processes with time constants shown in Equation (3.31), consistent with previous AFIT research [18, 14, 19]. All three are driven by zero-mean white Gaussian noise with strengths shown in Equation (3.34). The fourth SV-specific error source is due to inaccuracies in the clocks on board the SV's, and the final error source is based on line-of-sight errors between the SV's and the receiver. The model for these states is shown in

Equation (3.31),

$$\begin{pmatrix} \delta \dot{R}_{cl} \\ \delta \dot{R}_{trop} \\ \delta \dot{R}_{ion} \\ \delta \dot{R}_{Sclk} \\ \delta \dot{x}_{S_1} \\ \delta \dot{y}_{S_1} \\ \delta \dot{z}_{S_1} \end{pmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{500} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{1500} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \delta R_{cl} \\ \delta R_{trop} \\ \delta R_{ion} \\ \delta R_{Sclk} \\ \delta x_{S_1} \\ \delta y_{S_1} \\ \delta z_{S_1} \end{pmatrix} + \begin{pmatrix} w_{cl} \\ w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.31)$$

where the initial covariances for the states is given by;

$$P_{GPS}(t_0) = \begin{bmatrix} 0.25ft^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0ft^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0ft^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25ft^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25ft^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25ft^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25ft^2 \end{bmatrix} \quad (3.32)$$

and mean values and strengths of the dynamics driving noise are given by;

$$E\{w_{GPS}(t)\} = 0 \quad (3.33)$$

$$E\{w_{GPS}(t)w_{GPS}^T(t+\tau)\} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.004 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ft^2/sec \cdot \delta(\tau) \quad (3.34)$$

A quick reference of the truth model non-zero GPS and RRS dynamics matrix components is provided in Tables B.5 and B.5 of Appendix B. This ends the description of the 30-state truth model. Now the filter model will be presented.

3.3.3.2 The 2-State GPS Filter Model. Various research efforts have shown that two states provide a sufficient model for GPS [14, 6]. The primary argument is that the errors modeled by the other 28 states are small when compared to the two states common to all SV's. By adding dynamics driving noise, Q , and re-tuning the filter, the overall performance of the NRS can be maintained with the significantly reduced-order model of Equation 3.35:

$$\begin{Bmatrix} \delta \dot{R}_{clk_U} \\ \delta \dot{D}_{clk_U} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{clk_U} \\ \delta D_{clk_U} \end{Bmatrix} + \begin{Bmatrix} w_{R_{clk}} \\ w_{R_{clk}} \end{Bmatrix} \quad (3.35)$$

The values implemented for the dynamics driving noise strengths can be found in Table B.8 of Appendix E. It should be noted that in the tuning process, the measurement noise covariance values R have also been adjusted to achieve adequate tuning of the filter [9]. This completes the description of the GPS filter model. The next section presents the GPS measurement equations for both the truth and the filter models.

3.3.3.3 GPS Measurement Model. There are four GPS measurement updates, one for each of the satellite range signals received by the NRS filter. These measurement updates are once again difference measurements similar in structure to the RRS difference measurements. First the GPS truth model difference measurement will be fully presented, followed by a brief description of the filter measurement. The GPS difference measurement is formed by taking the difference of the INS-calculated pseudorange, PR_{INS} and the actual pseudorange, PR_{GPS} :

$$\delta z_{GPS} = PR_{INS} - PR_{GPS} \quad (3.36)$$

The real pseudorange, PR_{GPS} is the sum of the true range from the user to the satellite plus all the errors in the pseudorange signal propagation.

$$R_{GPS} = R_t + \delta R_{cl} + \delta R_{trop} + \delta R_{ion} + \delta R_{sclk} + \delta R_{uclk} + v \quad (3.37)$$

where

R_{GPS}	=	GPS pseudo range measurement, from SV to user
R_t	=	true range, from SV to user
δR_{cl}	=	range error due to code loop error
δR_{trop}	=	range error due to tropospheric delay
δR_{ion}	=	range error due to ionospheric delay
δR_{sclk}	=	range error due to SV clock error
δR_{uclk}	=	range error due to user clock error
v	=	zero-mean white Gaussian measurement noise

The second source of a range measurement is the INS itself, PR_{INS} . The derivation of PR_{INS} parallels that of R_{INS} from Section 3.3.2.3. PR_{INS} is the difference between the NRS-calculated position, X_U , and the satellite position from the ephemeris data X_S . This difference vector is represented below in the ECEF frame:

$$PR_{INS} = \left| X_U - X_S \right| = \left| \begin{Bmatrix} x_U \\ y_U \\ z_U \end{Bmatrix} - \begin{Bmatrix} x_S \\ y_S \\ z_S \end{Bmatrix} \right| \quad (3.38)$$

An equivalent form for Equation (3.38) is:

$$PR_{INS} = \sqrt{(x_U - x_S)^2 + (y_U - y_S)^2 + (z_U - z_S)^2} \quad (3.39)$$

Based on assumption 11 from Chapter I, Equation (3.39) can be approximated and rewritten in terms of the true range and a truncated first-order Taylor series, with perturbations representing the errors in X_U and X_S :

$$PR_{INS} = R_t + \left. \frac{\partial R_{INS}(X_S, X_U)}{\partial X_U} \right|_{(X_S, X_U)_{nom}} \cdot \delta X_U$$

$$+ \left. \frac{\partial R_{INS}(X_s, X_u)}{\partial X_s} \right|_{(X_s, X_u)_{nom}} \cdot \delta X_s \quad (3.40)$$

The solution for R_{INS} is found by substituting Equation (3.39) into Equation (3.40) and evaluating the partial derivatives to get:

$$\begin{aligned} R_{INS} = R_t &- \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_u \\ &+ \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_s + \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_s + \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_s \end{aligned} \quad (3.41)$$

Finally, the GPS pseudorange truth model *difference* measurement is given as:

$$\begin{aligned} \delta z_{GPS_t} &= R_{INS} - R_{GPS} \\ &= - \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_u \\ &\quad + \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_s + \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_s + \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_s \\ &\quad - [1] \delta R_{cl} - [1] \delta R_{trop} - [1] \delta R_{ion} \\ &\quad - [1] \delta R_{Sclk} - [1] \delta R_{Uclk} - v \end{aligned} \quad (3.42)$$

Similar to the RRS truth model measurements, the user position errors in Equation (3.42) can be transformed into the first three states of the filter or truth model using an orthogonal transformation [1].

The filter model for the GPS measurement updates can be derived in similar fashion as the RRS filter measurement updates. Since the filter model does not contain the states for the errors in the satellite position, these terms are removed from the equation. The filter model measurement equation can therefore be written as

$$\begin{aligned} \delta z_{GPS_f} &= R_{INS} - R_{GPS} \\ &= - \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_u \\ &\quad - [1] \delta R_{Uclk} - v \end{aligned} \quad (3.43)$$

The filter measurement noise strength, R will be tuned to attain adequate performance despite the reduction in order from the truth model and the Taylor series approximation. The measurement noise variances for both the filter and the truth model equations are provided in Table B.9 of Appendix B. This completes the description of the GPS measurement equations and the entire NRS filter and truth model equations. More detail can be found on these equations in past AFIT research [14, 18, 19]. The following sections will present the failure models implemented in this thesis.

3.4 Failure Models

The models explained previous to this section do not incorporate the occurrence of failures. This section explains how failures are added to the simulation, and clarifies the matching filter algorithms. This section begins with a restatement of the exact failures that are modelled. Next, the necessary changes to the truth model for generating simulated failures are presented. Finally the two matching filters are defined.

3.4.1 Description of the Failures. This section explains the mathematical models for the three types of failures: step bias, ramp offset, and increased measurement noise. The failures are restricted to single failures of a GPS satellite or RRS transponder range measurement. Doppler failures and altimeter failures are not considered in this thesis to maintain a reasonable number of multiple models in the simulation. Therefore only a single satellite or transponder measurement is affected during any one simulation run. Satellite 3 and transponder 1 have been chosen at random to be the signals altered by the step, ramp, and noise failures. The relative geometry of all the range measurements is presented in Chapter IV. Detailed mathematical definitions are now presented for these three failure types.

The step bias is modelled as a scalar increase in the range measurement of a satellite or a transponder. The bias will begin at time t_{fail} , and continue for a length of time,

t_{length} . Therefore, for all time, t_i , the step bias failure is defined as,

$$b(t_i) = \begin{cases} 0 & \text{if } t_i < t_{fail} \\ b_k & \text{if } t_{fail} \leq t_i \leq t_{fail} + t_{length} \\ 0 & \text{if } t_i > t_{fail} + t_{length} \end{cases} \quad (3.44)$$

where

$b(t_i)$	=	Step Bias Failure
b_k	=	Magnitude of the Step Bias
t_{fail}	=	Initial time of a failure
t_{length}	=	Time duration of the failure

The definition for the ramp offset is similar to that of the step bias. The ramp offset also begins at time t_{fail} and continues for duration t_{length} . Yet the ramp offset is a scalar magnitude that will actually increase during the duration of the failure. Equation (3.45) defines the ramp offset for all time t_i ,

$$r(t_i) = \begin{cases} 0 & \text{if } t_i < t_{fail} \\ r_k(t_i - t_{fail}) & \text{if } t_{fail} \leq t_i \leq t_{fail} + t_{length} \\ 0 & \text{if } t_i > t_{fail} + t_{length} \end{cases} \quad (3.45)$$

where

$r(t_i)$	=	Ramp Offset Failure
r_k	=	Slope of the Ramp Offset
t_{fail}	=	Initial time of a failure
t_{length}	=	Time duration of the failure

The increase in the measurement noise failure is inherently different from the other two failures. The noise failure increases the variance of the pre-existing Gaussian white measurement noise between times, t_{fail} and $t_{fail} + t_{length}$. Equation (3.46) shows how the failed

measurement noise term is changed to reflect the existence of the failure:

$$v_f(t_i) = \begin{cases} v(t_i) & \text{if } t_i < t_{fail} \\ n_k v(t_i) & \text{if } t_{fail} \leq t_i \leq t_{fail} + t_{length} \\ v(t_i) & \text{if } t_i > t_{fail} + t_{length} \end{cases} \quad (3.46)$$

where

$v_f(t_i)$	=	Measurement Noise for the Failed Satellite or Transponder Signal
$v(t_i)$	=	Scalar Measurement Noise, No Failure
n_k	=	Strength of the Measurement Noise Failure
t_{fail}	=	Initial time of a failure
t_{length}	=	Time duration of the failure

These three failure definitions define the scope of the failure analysis for this thesis. It is now necessary to integrate these failure definitions into the previously defined NRS truth model.

3.4.2 Failures in the Truth Measurement Equation. Up to this point, the truth model for the NRS filters has been defined without failure occurrences. To test the validity of the failure detection algorithms, it is necessary to alter the truth model so that failures will occur. To change the truth model, single failures are added to specific measurement updates over a predetermined window of time. For this work, all failures occur at time $t_i = 2000s$, and continue for $1000s$ ($s = seconds$).

Failures are added to the scalar measurement update equation for the failed measurement signal. Sections 3.3.3.3 and 3.3.2.3 provide the definitions for the scalar measurement equations for the satellite and transponder updates, assuming no failure has occurred. These equations are now redefined specifically for satellite 3 and transponder 1 to produce failures in the truth model. Equation (3.47) defines the new scalar measurement update equation for satellite 3:

$$\begin{aligned} \delta z_{GPS_{satellite}} = & - \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_u \\ & + \left[\frac{x_s - x_u}{|R_{INS}|} \right] \cdot \delta x_s + \left[\frac{y_s - y_u}{|R_{INS}|} \right] \cdot \delta y_s + \left[\frac{z_s - z_u}{|R_{INS}|} \right] \cdot \delta z_s \end{aligned}$$

$$\begin{aligned}
& - [1] \delta R_{cl} - [1] \delta R_{trop} - [1] \delta R_{ion} \\
& - [1] \delta R_{Sclk} - [1] \delta R_{Uclk} - v_f(t_i) + b(t_i) + r(t_i) \quad (3.47)
\end{aligned}$$

The new terms added to Equation (3.47), $b(t_i)$, $r(t_i)$, and $v_f(t_i)$, are defined in Equations (3.44), (3.45), and (3.46) respectively. Within the definitions of these three new variables, the magnitude of the failures is controlled by the scalar variables b_k , r_k , and n_k . The values used for simulation purposes in this thesis are presented in Table 3.48

The changes made to the scalar measurement equations for transponder 1 parallel those for satellite 3. The failed transponder equation is defined as,

$$\begin{aligned}
\delta z_{RRS_{Failed}} = & - \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_U - \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_U - \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_U \\
& + \left[\frac{x_T - x_U}{|R_{INS}|} \right] \cdot \delta x_T + \left[\frac{y_T - y_U}{|R_{INS}|} \right] \cdot \delta y_T + \left[\frac{z_T - z_U}{|R_{INS}|} \right] \cdot \delta z_T \\
& - [1] \delta R_{atm} - [1] \delta R_b - v_f(t_i) + b(t_i) + r(t_i) \quad (3.48)
\end{aligned}$$

As before, the failure variables added to the above equation are defined in Equations (3.44), (3.45), and (3.46). Once again Table 3.2 contains the failure magnitude values, b_k , r_k , and n_k , for each of the simulation runs. The magnitudes of the failures are based upon an analysis

Table 3.2 Definition of Failure Variables for each Simulation Run

Run	Failure Type	Failed Measurement	b_k	r_k	n_k	t_i	t_{engin}
1	Baseline	None	0	0	1	N/A	N/A
2	Step Bias	Transponder 1	800	0	1	2000	1000
3	Ramp Bias	Transponder 1	0	1	1	2000	1000
4	Noise Increase	Transponder 1	0	0	10	2000	1000
5	Step Bias	Satellite 3	3500	0	1	2000	1000
6	Ramp Bias	Satellite 3	0	4	1	2000	1000
7	Noise Increase	Satellite 3	0	0	15	2000	1000

of the impact of the failures on the filter residuals. The failure magnitudes displayed in Table 3.2 are chosen high enough so that the impact of the failures on the residuals can be clearly distinguished from failure-free system dynamics, yet low enough not to induce instantaneous instabilities in the filter state propagation.

The changes defined in this section are the only changes made to the truth model. The changes produce single failures (step, ramp, and noise) in either a single transponder or satellite difference measurement.

3.4.3 The GLR Failure Models. As defined in Section 2.3.2.2, the GLR algorithm has been modified to become a simple matching filter. This new function, defined as $\Upsilon(t_i)$ in Equation (2.58), is referred to as the modified GLR matching filter. The EKF produces almost all the necessary variables to compute $\Upsilon(t_i)$ in Equation (2.58). However there are three variables that need to be defined to complete the matching filter description. First the Chi-Square test must provide the identity of the failed measurement. This allows the matching filter to predefine the vector $d(t_i)$ as $d_x(t_i)$. Therefore $d_x(t_i)$ will be defined as,

$$\begin{aligned} d_x(t_i) &= \begin{bmatrix} SV1 & SV2 & SV3 & SV4 & TR1 & TR2 & TR3 & TR4 & TR5 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (3.49)$$

for the satellite failure runs and,

$$\begin{aligned} d_x(t_i) &= \begin{bmatrix} SV1 & SV2 & SV3 & SV4 & TR1 & TR2 & TR3 & TR4 & TR5 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (3.50)$$

for transponder failure runs.

The other two variables that must be defined for the GLR matching filter are the step and the ramp failure models. These models are closely related to the definitions of the failures in the truth model (Equations (3.44) and (3.45)). The GLR failure model for the step failure is defined as,

$$n_{step}(t_i) = \begin{cases} 0 & \text{if } t_i < \hat{\theta}_x \\ 1 & \text{if } t_i > \hat{\theta}_x \end{cases} \quad (3.51)$$

The GLR failure model for the ramp failure is derived similarly to the step failure model. Equation (3.52) gives the definition for the GLR ramp failure model:

$$n_{ramp}(t_i) = \begin{cases} 0 & \text{if } t_i < \hat{\theta}_\chi \\ (t_i - \hat{\theta}_\chi) & \text{if } t_i > \hat{\theta}_\chi \end{cases} \quad (3.52)$$

For both the step and the ramp GLR failure models, unity magnitude has been declared. The GLR matching filter is merely attempting to determine the type of failure, not its magnitude.

The estimate of the failure magnitude is calculated from Equation (2.60). This equation is rewritten below,

$$\hat{\nu} = \frac{\Psi(t_i, \theta)}{S(t_i, \theta)} \quad (3.53)$$

To estimate the magnitude of the step and the ramp failures, all the information available needs to be incorporated into this expression. Therefore using the known quantities, $d_\chi(t_i)$, $\hat{\theta}_\chi$, $n_{step}(t_i)$, and $n_{ramp}(t_i)$, the step and ramp magnitude estimates are defined as:

$$\hat{\nu}_{step} = \frac{\Psi_{step}(t_i, \hat{\theta}_\chi)}{S_{step}(t_i, \hat{\theta}_\chi)} \quad (3.54)$$

for the estimate of the step failure and:

$$\hat{\nu}_{ramp} = \frac{\Psi_{ramp}(t_i, \hat{\theta}_\chi)}{S_{ramp}(t_i, \hat{\theta}_\chi)} \quad (3.55)$$

for the estimate of the ramp failure. Depending upon the outcome of the modified GLR matching filter, one of these two estimates will provide an estimate of the step or ramp magnitude.

The GLR matching filter algorithm has now been completely defined. The following section defines a second matching filter algorithm. In the end, the results of these two matching filters will be analyzed to determine which provides better failure type identification for the MNRS algorithm.

3.4.4 *The CSPR Matching Filter Models.* The CSPR algorithm identifies failure type by examining the output of the Chi-Square test. The Chi-Square test results will display specific characteristics that can specify the failure as a step, ramp, or noise failure. In fact, the bias detected in the Chi-Square output when a failure occurs will generally match the shape of the failure itself. Figure 3.3 shows the effect in the Chi-Square test that will identify the three different types of failures. While the shape of the ramp and noise failures correspond to the exact shape of their failures, the Chi-Square test will tend to recover from a step bias failure. Therefore the CSPR filter will expect a slightly negative slope on a step bias failure. While a human can merely look at the output of the

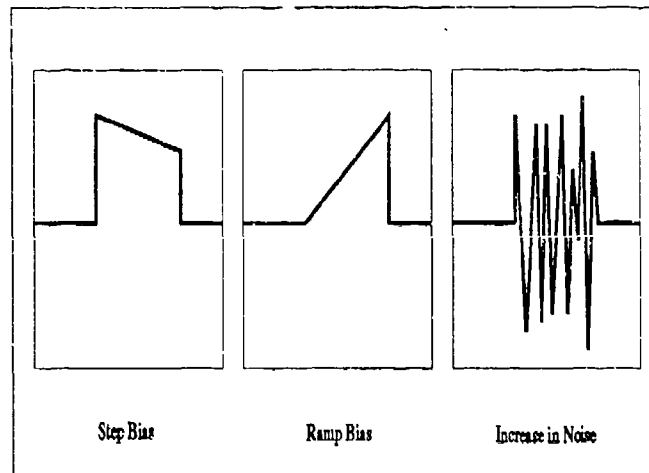


Figure 3.3 Shape Chi-Square Test as a Result of the Three Failure Types

Chi-Square test to determine the failure type, quantitative criteria need to be established to differentiate between the types of failures. The criteria chosen are the slope and the intercept of a first order line fit of the Chi-Square test over the period of the failure. The slope and intercept of the line fit are *distinct* for the three types of failures. The line fit algorithm has been adopted from MATLAB [8]. The exact points interpolated will be those that violate the Chi-Square failure threshold. Table 3.3 explains how the slope and intercept criteria distinguish between the three failures.

The CSPR matching filter assumes, like the GLR, that the failed measurement has already been identified. The CSPR algorithm estimates the type of failure for all time after the Chi-Square test has exceeded the failure threshold at time, $\hat{\theta}_x$. It should be

Table 3.3 Slope and Intercept Values for the Three Types of Failures

Type of Failure	Slope	Intercept
Additive Step Bias	Zero or Slightly Negative	Large and Positive
Additive Ramp Bias	Positive	Zero or Slightly Negative
Increase in Meas. Noise	Zero Mean over Time	Zero Mean over Time

noted that the estimate of failure type will improve over time as the first order line fit of the Chi-Square output better matches the shape of the failure. The results of the CSPR matching filter are compared against the GLR matching filter in Section 4.5.

3.4.5 Choosing between the NRS Filters. This section defines the exact algorithm that will be used to select the filter both when a failure has and has not occurred. The algorithm uses the results of the bank of Chi-Square tests and the relative quality of the satellite set's geometries to determine which filter is yielding accurate results. The criteria implemented to rate satellite geometry is the Position Dilution of Precision(PDOP). First the no failure selection algorithm is discussed, followed by an explanation of what happens when a failure occurs.

For all the Chi-Square tests, a threshold value has been established. As long as all the filters are below the threshold, no failure has been declared, and the MNRS is operating in the no failure condition. When no failure has occurred, the NRS filter 1 is used for navigation, because the filter has the best possible combination of satellites and transponders for navigation (assuming that only four satellites and five transponders can be used at a time). This navigation solution is the best since it uses the four satellites with the best PDOP, and the five primary transponders. Ideally it would be beneficial to have an eleventh filter updating with all the possible satellite and transponder measurements. Theoretically, this filter would have the best navigation correction. The eleventh filter has not been implemented in this work due to the computational load of simulating another NRS filter.

A failure is declared when the Chi-Square test for NRS filter 1 exceeds its threshold. This means that the best navigation solution, filter 1, has been distorted with a failure. At this point only one of the other filters should have a Chi-Square test that has not exceeded its threshold. This unaffected filter is now used to correct the INS navigation solution. The time at which the NRS filter 1 *declares* a failure becomes the Chi-Square estimate of the time of failure, $\hat{\theta}_\chi$. It should be noted that this estimate has varying degrees of accuracy depending on the type of failure that has affected the system. Equation (3.56) expresses the relationship between the actual time of failure and the Chi-Square estimated time of failure:

$$t_{failtime} = \hat{\theta}_{\chi} - t_{delay} \quad (3.56)$$

The *unknown* quantity, t_{delay} , varies dramatically depending on whether the failure is a step bias, ramp offset, or a noise increase. In fact, the delay in detecting the ramp offset can seriously impact the results of the matching filter algorithms. Chapter IV addresses this issue of detection delay in the MNRS algorithm. However for the purposes of this research, $\hat{\theta}_\chi$ is considered the best estimate of the time of failure.

Thus the estimated time of failure and the identity of the failed measurement are passed to the matching filter algorithms. This allows the matching filter algorithms to identify the type of failure. Chapter IV will provide more insight into these algorithms with the results of the various failure runs.

3.5 Chapter Summary

This chapter has presented the details for both the navigation filter and failure models. The basis for the measurement models has been discussed to help describe the intricacies of the NRS design. The state and dynamics model descriptions illustrate the high degree of nonlinearity and time-variance of the system. The reduced-order filter models have also been presented. The methods used to induce the failures in the simulations have been shown, along with the models for the matching filters designed to detect and isolate these failures. Results and analysis of these simulations are presented in the next chapter.

IV. Results and Analysis

This chapter examines the results of the MNRS FDIR algorithm. First the chapter presents flight simulation data to help define the bounds of this thesis study. Next, the tuning of the NRS filter is analyzed with regard to filter performance in a failure-free environment. Then the effects of different failures are shown on a single NRS filter. Finally, the results achieved with the MNRS FDIR algorithm are presented in full detail, including a comparison of matching filter algorithms.

4.1 The Simulation Specifications

This section presents the specifications of the simulation. The specifications are defined as the flight profile of the aircraft, the location of the transponders, and the position of the satellites during the simulation.

All the simulation runs in this study use a single flight profile generated by PROF-GEN. The flight profile has been derived from a two-hour flight of a fighter aircraft. Due to the computational load of running a multiple model simulation, only the first 4000 seconds of the 7200 second flight profile are used in the simulation studies. Figure 4.1 provides a three dimensional rendition of the flight profile. The exact flight path is shown as the aircraft climbs to altitude before proceeding through a high speed low altitude mission. The time of failure onset is also displayed in Figure 4.1. This failure onset time of 2000 seconds is the same for all failure simulation runs. This failure onset time is arbitrarily chosen at the mid-point of the flight profile.

The flight profile was chosen over a region of the CIGTF RRS test range to allow the transponders to be implemented in the integrated navigation system. While the test range has a multitude of transponders, the number of transponders used in this thesis has been limited to six; five used for the measurement updates to any one NRS filter. The geographic coordinates of the six transponders sights are presented in Table 4.1. Figure 4.2 shows how the locations of the transponders correspond to the aircraft's position during the flight profile. Despite the proximity of the aircraft to the transponder range, transponder

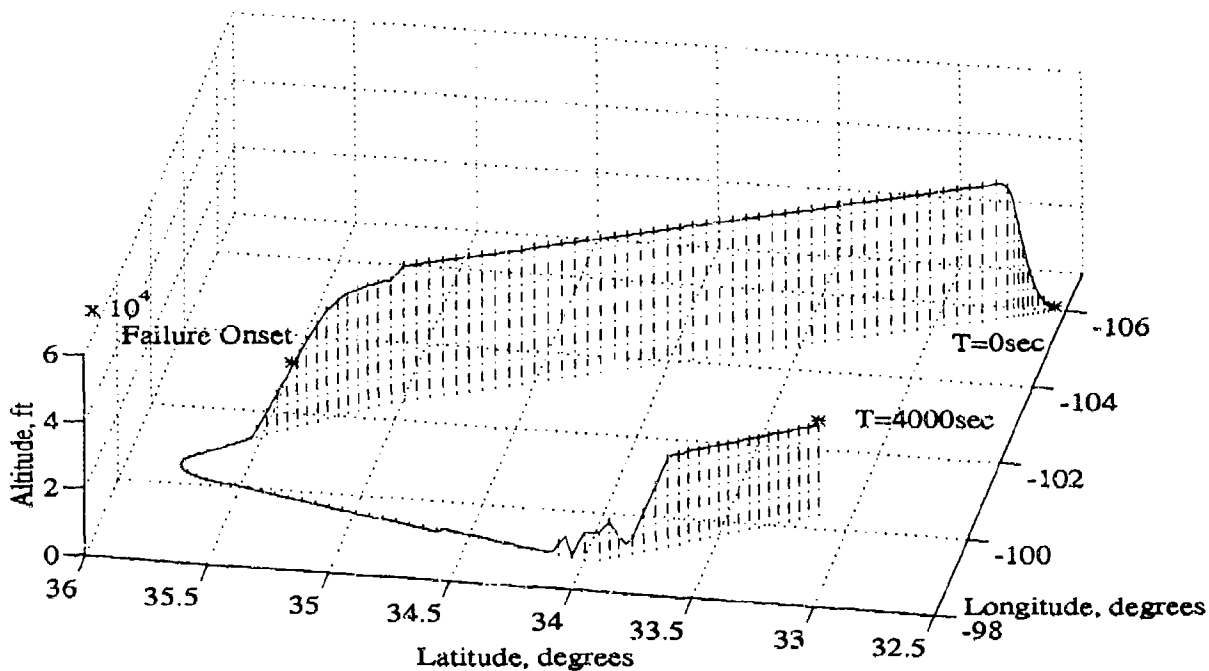


Figure 4.1 Flight Profile used for the MNRS Simulations

geometry has little impact on the quality of the navigation solution. Section 4.4 discusses which geometric considerations do impact the results of the MNRS algorithm.

Along with the transponders, the aircraft also receives measurement updates from GPS satellites. Table 4.2 gives the latitude and longitude of the five satellites used in this simulation at both the beginning and end of the simulation. Figure 4.3 shows the relative positions of the satellites during the simulation. In Figure 4.3, the orbit path of

#	Name	Latitude	Longitude	Altitude
1	Tula PK, NM	33.01°36'	-106.08°20'	1322.5272 <i>ft</i>
2	TDC, NM	32.55°58'	-106.08°50'	1241.7552 <i>ft</i>
3	Oscura Pk, NM	33.44°58'	-106.22°14'	2417.5144 <i>ft</i>
4	Salinas, NM	33.17°55'	-106.31°44'	2695.11 <i>ft</i>
5	Sac Peak, NM	32.47°16'	-105.49°15'	2804.81 <i>ft</i>
6	Twin Buttes, NM	32.42°12'	-106.07°38'	1365.71 <i>ft</i>

Table 4.1 Location of the Transponders Transmitters in the MNRS Simulation

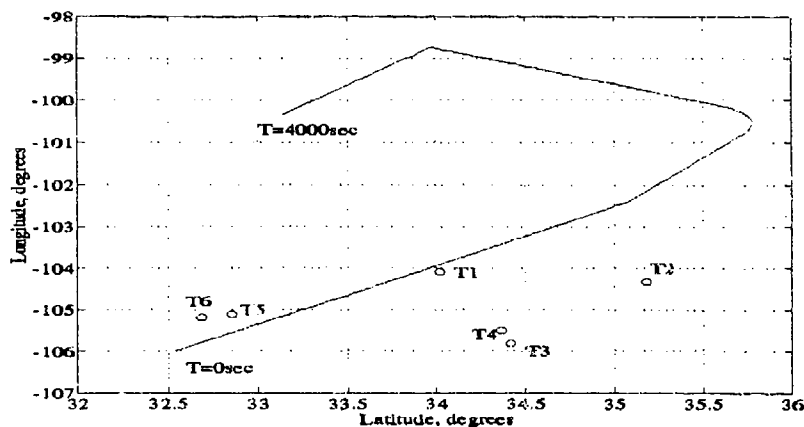


Figure 4.2 Location of the Transponder Transmitters with Respect to the Flight Profile

Satellite	Initial Latitude	Initial Longitude	Final Latitude	Final Longitude
1	67.90°	-172.21°	58.63°	42.30°
2	34.75°	-73.16°	38.29°	-42.78°
3	-43.03°	-129.49°	18.18°	-122.89°
4	78.43°	-49.15°	-14.42°	-56.58°
5	33.82°	-150.69°	40.09°	176.34°

Table 4.2 Location of the GPS Satellites during the MNRS Simulation

the satellites is traced up to the final position of the satellites at time $T = 4000$ seconds. The dashed lines in Figure 4.3 represent the range vector at time $T = 4000$ seconds between the five satellites and the aircraft. Despite the movement of the satellites, all five satellites are in view throughout the flight profile of the aircraft. The five satellites have been chosen to prevent satellite switching during the simulation runs.

The purpose of this section has been to provide credibility for the simulation results. A worthwhile computer simulation must demonstrate parallels to the real world. The real world parallels for this thesis are the real world GPS satellite positions, the RRS transponder positions, and the flight profile. While the navigation information provided to the NRS is simulated, it is a reasonable model for real world navigation data. The next section presents the filter tuning results of a single NRS filter within the simulated environment.

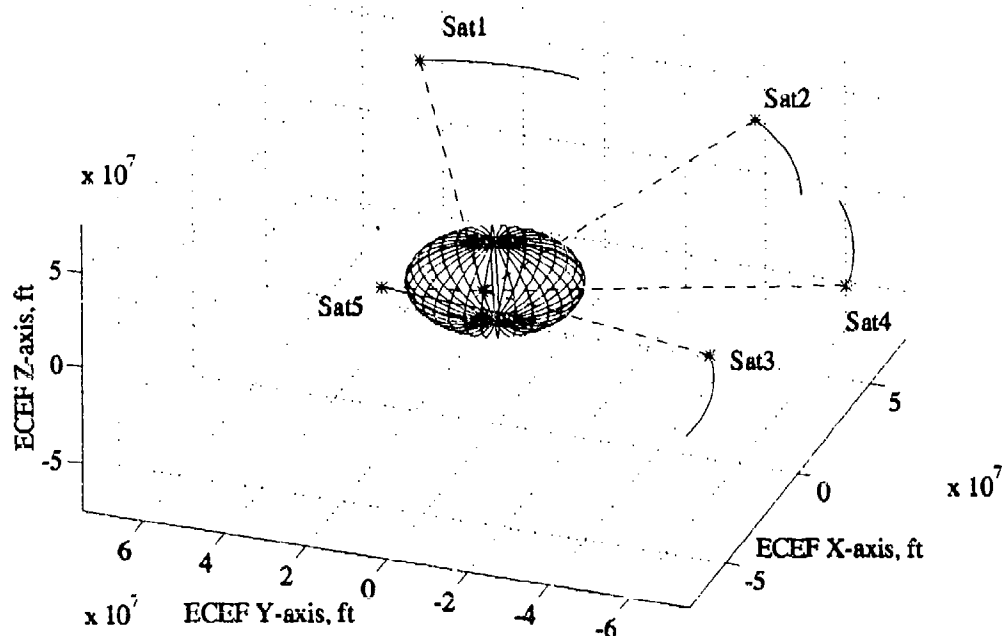


Figure 4.3 Satellites Orbit Profiles with Respect to the Flight Profile

4.2 Performance of the NRS Filter

This section documents the performance of the NRS filters in a failure free environment. This section presents results from NRS filter 1 simulations. This section begins with an overview of the changes made to the NRS filter during this thesis research. Next, criterion is established to evaluate the accuracy of the NRS filter tuning. Once criterion has been established, the results of the NRS filter tuning are compared with past AFIT results and specifications provided by CIGTF. Finally, the EKF tuning process is addressed with an analysis of two trade-off decisions made in the tuning of the NRS filter.

The research for this thesis began in earnest with the blending of the work done by the Avionics Directorate[3] and past AFIT research[19] into a new NRS computer model. While this process required a great deal of time and effort, only two theoretical changes were made to the NRS algorithm. The first change is within the truth model of the INS.

Previously, little information was known about the barometric altimeter error states. The model originates from the Litton documentation [4]. However research by the Avionics Directorate uncovered inconsistencies and problems within the Litton documentation. These errors were corrected by the Avionics Directorate with the help of Litton [3]. While the corrections do not dramatically alter the results of the research, the changes do improve the accuracy of the NRS truth model. The second problem uncovered in the NRS model existed in the GPS truth model states. Past AFIT work failed to include dynamics driving noise to the following GPS truth model error states; tropospheric delay, ionospheric delay, and code loop range errors. This oversight allowed past NRS filters to demonstrate unusually good performance. By correcting the GPS and barometric altimeter error states, the truth model becomes a more credible bridge between computer simulation and the real world. Without a highly accurate truth model, algorithms developed in computer simulation have little credibility in the real world. Due to the quality of support provided by the Avionics Directorate and past AFIT research, the remaining model development problems were easily overcome, and the NRS tuning process was able to proceed on schedule.

To ensure accurate navigation performance, two tuning criteria are used to evaluate the filter's state variable performance. The first criterion is the ability of the fifteen filter state variables to track the corresponding fifteen truth state variables adequately. Optimally, the Monte Carlo average of the difference between the filter and truth state variables should be zero mean over the entire profile. The second tuning criterion is the accuracy of the filter's standard deviation estimate. The filter-predicted zero \pm the standard deviation of each state, $0 \pm \sigma_f$, should match the Monte Carlo computed standard deviation of the difference between filter and truth states, $\bar{e}_{m.c.} \pm \sigma_{m.c.}$. Both tuning criteria can be plotted on a single graph. Appendix C provides the mathematical description of all the variables used in each of the filter tuning plots.

Before the filter tuning plots are presented, a variable transformation used in this thesis must be defined. While the first nine filter states are stored and propagated in a computed true reference frame, this frame of reference does not provide a great deal of physical insight. Therefore results of the filter tuning are transformed to the East-North-Up (ENU) frame; hence, position errors are presented in latitude, longitude, and altitude

error. The velocity and tilt errors are similarly documented in the East-North-Up frame. The transformation used to present the material in the ENU frame is orthogonal and has been documented in past AFIT research [14, 19].

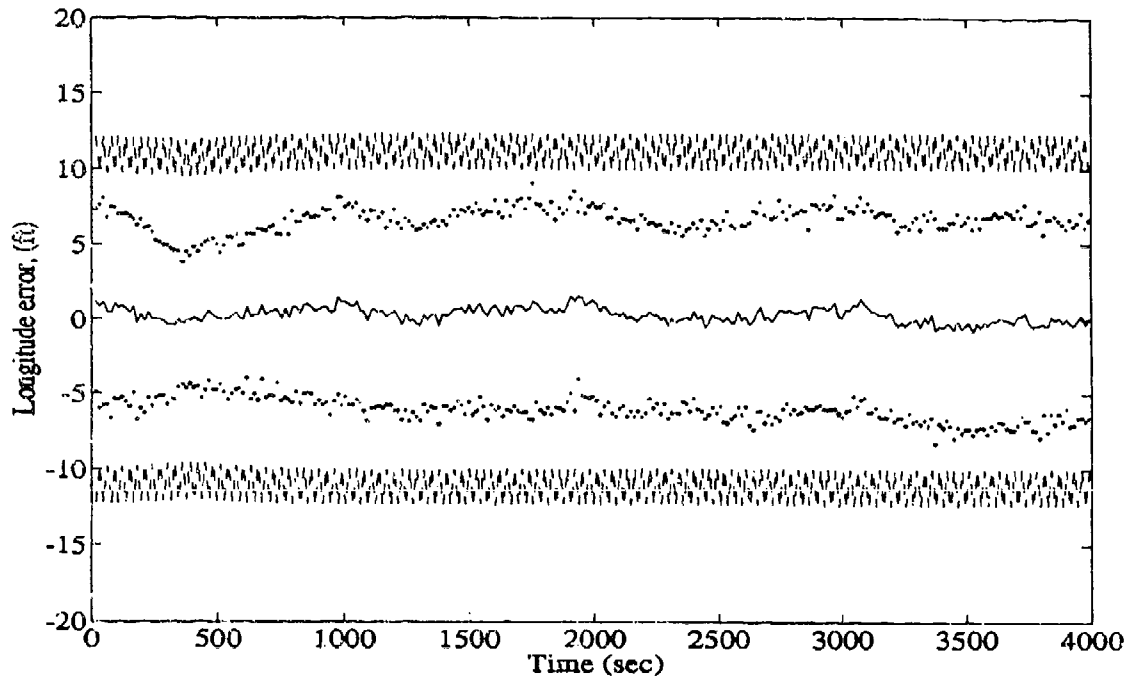


Figure 4.4 Longitude Filter Error Tuning Plot

To show the results of the NRS filter tuning process, the longitude error tuning plot is analyzed in Figure 4.4. This figure allows the two tuning criteria to be evaluated at the same time. Within Figure 4.4, the *Monte Carlo* calculated mean value of the difference between the filter and truth models longitude error is displayed by the solid line tracking the x-axis. The filter's ability to track the longitude error of the truth model is demonstrated by the zero mean nature of this trace. The second tuning criterion is evaluated by examining the two pair of symmetric traces in Figure 4.4. The pair of dotted lines is the Monte Carlo calculated mean \pm standard deviation of the difference between the filter and truth model longitude error. The dashed lines represent zero \pm the filter-predicted standard deviation of the error. Optimal filter tuning would have these two pair of lines overlapping over time. The pair of lines in Figure 4.4 are close to overlapping with

the filter-predicted standard deviation greater than the Monte Carlo calculated (i.e., the filter is less confidence in itself than it could be). The conservative nature of this tuning proved to be necessary to maintain good tracking of the most important navigation states. Filter tuning plots for all the filter states are located in Appendix C.

Table 4.3 Temporal Averages of True Filter Errors (1σ)

Filter	Latitude (ft)	Longitude (ft)	Altitude (ft)	East Vel (fps)	North Vel (fps)	Up Vel (fps)	East Tilt (arcs)	North Tilt (arcs)	Azimuth (arcs)
Desired	13.00	13.00	40.00	0.100	0.100	0.400	n/a	n/a	n/a
NRS 1992	4.00	7.22	8.63	0.032	0.029	0.070	3.13	4.49	17.57
NRS 1993	3.45	6.64	15.96	0.062	0.062	.137	4.05	4.41	27.94

Despite the best possible filter tuning, the actual quality of the filter performance is also limited by the size and the structure of the model itself. Therefore *good* filter tuning cannot be the final criterion used to evaluate a filter. The quality of a navigation filter is best described by the errors made in the estimation of position, velocity, and orientation. Table 4.3 contains the temporal averages for these errors over the complete flight profile. The table provides two references for comparison; the original CIGTF specifications for the system and the results attained from a fifteen-state NRS filter implemented in past AFIT research [19]. While the performance of the filter easily meets the CIGTF specifications, the results are slightly degraded with respect to past AFIT research. The slight degradation in performance of some states is attributed to two factors. First, the model last year did not employ dynamics driving noise on all the GPS truth states. Second, the new NRS model uses only five transponder signals for measurement updates while past AFIT research used six. Therefore, given the results of the NRS tuning process, the filter achieved is adequate for the purposes of simulating the MNSS FDIR algorithm.

To understand the mechanics of filter tuning better, background is provided on two specific tuning decisions. The first tuning decision is the choice of tuning criterion. If residual performance is more important than state variable performance, the residuals can be used instead of state variables to evaluate filter tuning. In this case, the filter-predicted (zero-)mean \pm sigma of the residuals is compared to the actual Monte Carlo calculated

mean \pm sigma of the residuals. The standards for good performance are the same as for the state variable tuning. An example of a residual plot for NRS filter 1 is given in Figure 4.5. This figure clearly shows that the residuals are zero-mean and appear uncorrelated over time, yet highly conservative with regard to its estimate of residual standard deviation. While the conservative nature of the residuals impedes FDIR, the importance of state variable performance is deemed to outweigh FDIR concerns. Degradation of the navigation quality for the sake of FDIR performance is not acceptable. Therefore, due to this constraint, a trade-off has been made in favor of state variable estimation over residual performance.

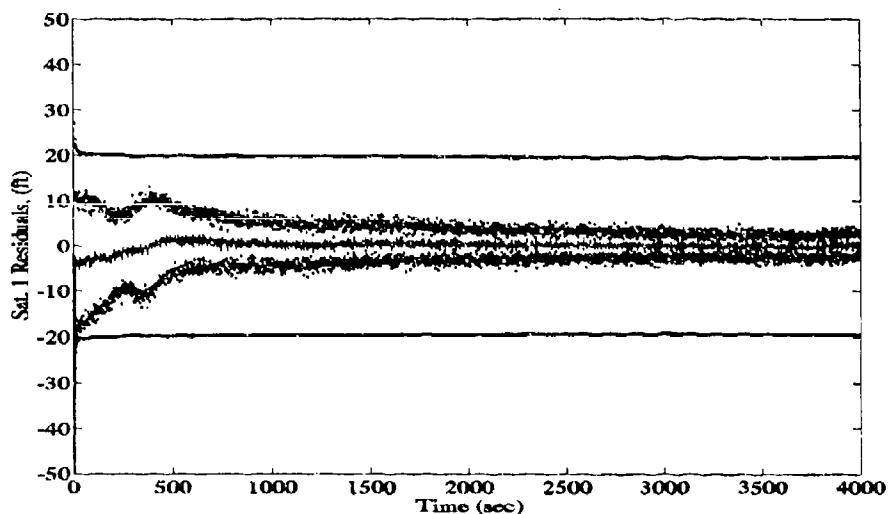


Figure 4.5 Example of Scalar Residual Tuning of the NRS Model

The second tuning decision that is addressed, regards the relative quality of tuning between individual state variables. Quite often tuning quality on one state must be sacrificed to achieve the desired performance of a more important state. Figure 4.6 shows the two coupled GPS states in the filter, GPS user clock bias and drift. The user clock bias shows excellent performance under both tuning criteria, while the drift appears to be tuned far too conservatively. These two states are *coupled*, so changes to the user GPS clock drift also affect the user clock bias state. Attempts to improve the confidence of the drift state seriously degraded the tracking of the user clock bias state. Since the accuracy of the user

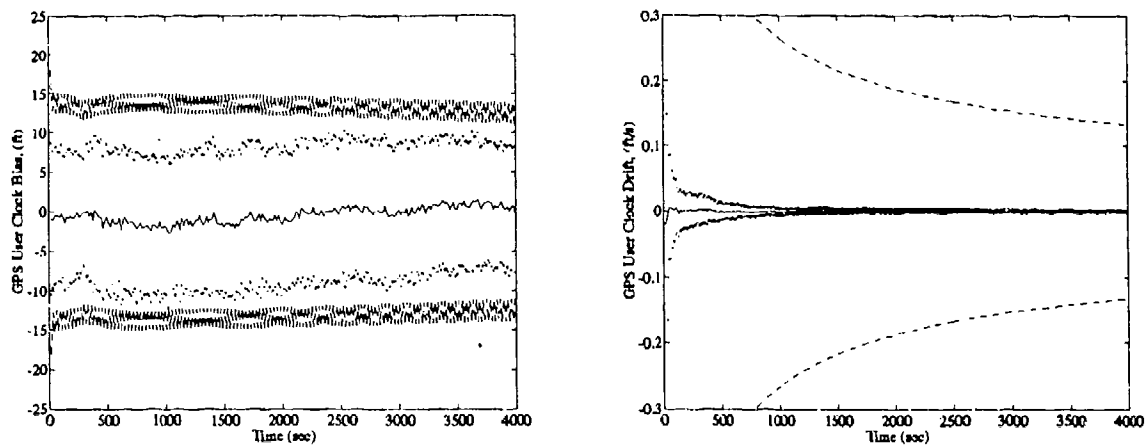


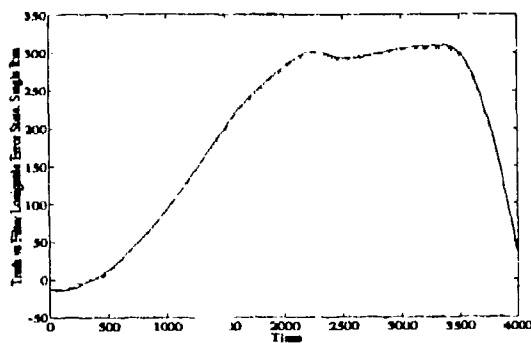
Figure 4.6 Relative Tuning of the GPS User Clock Bias and Drift States

clock bias state is more important, the drift state has been left with its conservative tuning. While this tuning is acceptable for the needs of this work, had GPS drift been more crucial to the operation of the overall system, the tuning would need to be reevaluated.

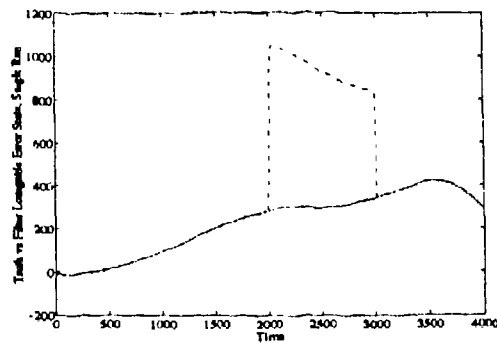
By describing a few of the tuning decisions made in this thesis, hopefully the subjective nature of filter tuning has been established. *Good* filter tuning can change depending on the situation and the stipulations placed upon the designer. The NRS filter has been tuned to attain the best estimates for position, velocity, and attitude error. If the filter is applied to other new applications in the future, the tuning criteria and results should be reconsidered. This concludes the discussion on filter tuning; the next section presents the results from the MNRS FDIR simulation runs.

4.3 Performance of the MNRS Filter

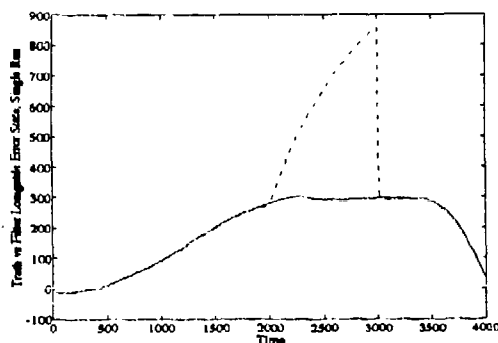
The MNRS simulation runs provide definitive results in favor of multiple model FDIR algorithms. The presentation of the MNRS results is divided into three sections. First the effect of the failures on a single NRS filter is shown. These results are included to motivate the need for a *recoverable* failure analysis algorithm. Once the need of for the MNRS has been established, the results from the MNRS simulation runs are presented. Finally the impact of satellite geometry on individual NRS filters is analyzed in the last section.



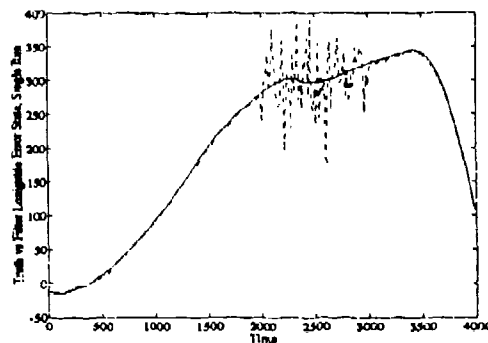
(a) No Failure



(c) Step Bias



(b) Ramp Offset



(d) Noise Increase

Figure 4.7 Comparison of the Effect of Satellite Failures on the Longitude Error State Tracking of NRS Filter 1

4.3.1 *Impact of the Failures on the NRS Navigation Solution.* To justify the need for an FDIR algorithm, the effect of the failures on the NRS filter are shown. To analyze the effect, a single run of the filter-estimated longitude error is compared to the failure-free truth longitude error. When the filter is operating in a failure-free environment, the filter longitude error can track the truth model error, as seen in Figure 4.7.a. Each of the three types of failures added to the truth model range measurements, bias the filter model such that it can no longer track the truth model state variables. Figures 4.7.b, 4.7.c, and 4.7.d show that the filter solution for longitude error deviates from the truth model. These plots have been included to motivate the need for a *recoverable* system. The knowledge of the existence of a failure alone does not aid the user in his navigation. Without recovery, all that can be done is to turn off the filter corrections to the INS. Yet with the MNRS

recoverable filter algorithm, the INS receives accurate navigation corrections throughout the duration of the failure.

4.3.2 Results of the MNRS FDIR Algorithm. The MNRS FDIR algorithm results are discussed in this section. Seven failure runs are documented: six failure runs and a baseline run for comparison. For each of these seven runs, all ten NRS filters have been simulated running in parallel over the 4000-second flight profile. Each of the NRS filters is updated with identical measurement signals every two seconds over a 15-run Monte Carlo simulation. While past AFIT research implemented only 10-run Monte Carlo analysis, the increased computing capacity of AFIT has allowed 15-run Monte Carlo analysis to be implemented. This increase in the number of simulation runs improves the validity of the Monte Carlo calculated mean and standard deviation of the state variables and residuals. The analysis of the simulations not only focuses on the effectiveness of the MNRS as an FDIR algorithm, but also on the different failure runs with regard to residual performance.

The majority of the MNRS results are contained in the Appendices D-J. Appendices D, E, and H contain Monte Carlo scalar residual data for each measurement update of each NRS filter in each failure run. Appendices F, G, I, and J contain Monte Carlo scalar residual data for only one measurement update of each NRS filter in each failure run. The one residual plot included is for the failed measurement. These single residual plots have been included to allow the reader to see the actual impact of the failure on the performance of the filter. Appendix K contains the Chi-Square test results of each NRS filter in each failure run. Appendix K also includes a chart for each run, displaying which filter is actively providing the navigation correction at each position time. The raw data collected in this thesis has been included in these appendices, not to overwhelm the reader, but to allow a chance to evaluate the conclusions reached by the author. Without access to the raw simulation data, analysis of the analyst can be difficult.

4.3.2.1 Baseline, No Failure. To provide a comparison with the failure runs, one complete MNRS run is conducted with no failures induced in the truth model. Individual scalar residual and the Chi-Square test results are analyzed to verify the operation of the ten NRS filters and to assist in the choice of a failure detection threshold.

A 15-run Monte Carlo simulation demonstrates the good residual performance of all the NRS filters. As can be seen in Appendix D, the Monte Carlo calculated scalar residuals are zero-mean over time. Looking specifically at NRS filter 1 (Figures D.1 and D.2), the scalar residuals reflect conservative tuning with regard to measurement confidence. This conservative tuning with regard to the residual covariance, $[HP^+H^T + R]$, removes high frequency effects of the scalar measurement updates: both unwanted noise and measurement information. The navigation information lost due to the conservative tuning is partially recovered by the high sampling rate of the measurements. The two-second sampling rate used for all the measurement updates is equal to or faster than any implemented in past AFIT research [14, 19]. Therefore the effect of the reduced confidence in the measurement updates is offset by the high sampling rate.

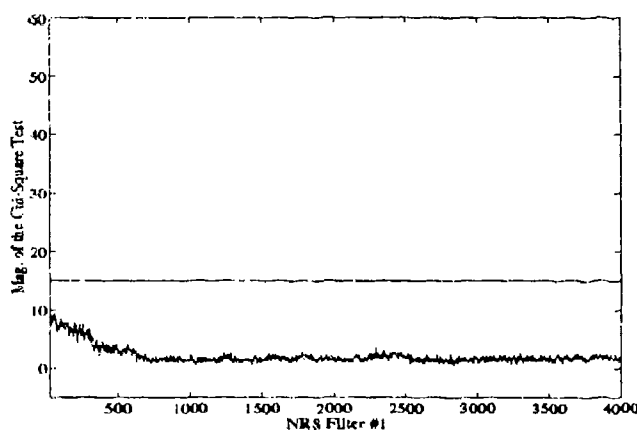


Figure 4.8 Baseline Chi-Square Test Results with the Chosen Failure Threshold=15

The baseline simulation also provides the necessary data to select the threshold for the Chi-Square test. Threshold selection, as discussed in Section 2.3.4, is dependent on the failure-free response of the residuals to normal system dynamics. The threshold must be chosen to maintain a low false alarm rate in a failure-free environment. Figure 4.8 shows the results of the Chi-Square test conducted on NRS filter 1. The threshold has been chosen greater than the maximum value of the Chi-Square test. This conservative choice of threshold eliminates the possibility of a false alarm for *the purposes of this simulation*. This simulation has been limited to the flight of one flight profile under 15 distinct represen-

tations of dynamics and measurement truth model noise. More comprehensive simulation testing and hardware testing would be subsequent steps to iterate the magnitude of the threshold for the Chi-Square test further.

Since no failure has been introduced to the system for this failure run, none of the NRS filters violate the threshold test during the simulation, as seen in Figures K.1 and K.2. Therefore, by the criteria established in Section 3.4.5, the NRS filter 1 will continue to provide the error state corrections to the INS navigation solution. Figure K.3 graphically displays which of the NRS filters provides the navigation correction to the INS throughout the simulation. With the baseline residual performance verified and the Chi-Square threshold established in a failure-free simulation run, the response of the MNRS algorithm to the six failures is presented next.

4.3.2.2 Transponder Bias Failure. The transponder step bias failure run differs from the baseline run in that a base of 800ft is added to the Transponder 1 range signal between 2000-3000 seconds in the simulation. During the simulation, the Chi-Square test exceeds its threshold in each filter that receives the failed measurement signal. Figures K.4 and K.5 show that only NRS filter 10 does not exceed the threshold. Therefore, the MNRS filter switches from NRS filter 1 to 10 for the duration of the threshold test violation on NRS filter 1 (see Figure K.6). Appendix E contains the individual scalar residual plots for each filter for this simulation to support these results.

While the MNRS FDIR technique adequately performs during the failure run, the residual response to the failures is somewhat unexpected. Past research of EKF FDI techniques has shown that the residuals of a EKF tend to recover from a step failure over time [2, 5]. The recovery from the failure is caused by the consistent nature of the step failure. Slowly the filter will compensate for the existence of the step bias, and the residuals will return to zero mean. When this occurs, the Chi-Square test will no longer signal a failure, and an observer might conclude that the bias is no longer affecting the system. This type of residual behavior does not allow for prolonged detection of additive constant biases failures in the measurement signals.

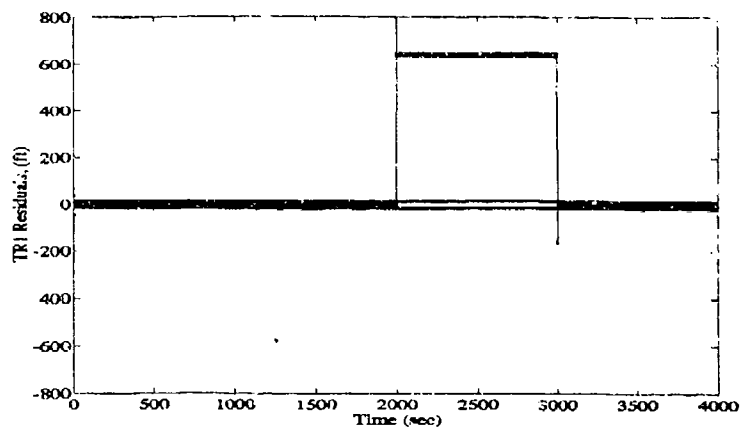


Figure 4.9 For NRS Filter 1, the Transponder 1 Measurement Residual Response to a Transponder Step Bias=800ft

In contrast to the expected residual response to a step bias, the scalar residual of the Transponder 1 measurement does not recover from the failure over time. As can be seen in Figure 4.9, the residual bias is constant during the failure time. The lack of recovery is caused by the modelling of the transponder error states in the NRS filter. Since both transponder state variables are modelled as random biases with relatively small levels of dynamics driving noise, the filter has no state variables that can easily *absorb* the failure offsets. The magnitude of the tuning noise covariance would need to be dramatically increased to absorb unmodelled biases. However increasing this tuning noise would greatly reduce the filter's state variable performance. Therefore it is the choice of filter model and the relative tuning strengths that causes the filter to maintain detection over the entire lifetime of the failure.

4.3.2.3 Transponder Ramp Failure. As with the transponder step bias, the ramp failure of transponder signal is easily detected by the MNRS FDIR algorithm. As seen in Figures K.7 and K.8, each filter receiving the failed measurement rapidly exceeds the threshold of the Chi-Square test. Figure K.9 shows the effective switching of filters for the duration of the threshold violation. For the transponder ramp failure, the threshold violation directly corresponds to the duration of the measurement failure. Therefore the

MNRS FDIR algorithm both detects and recovers from the transponder ramp failures. Appendix F contains the transponder 1 scalar residual plots for all ten NRS filters.

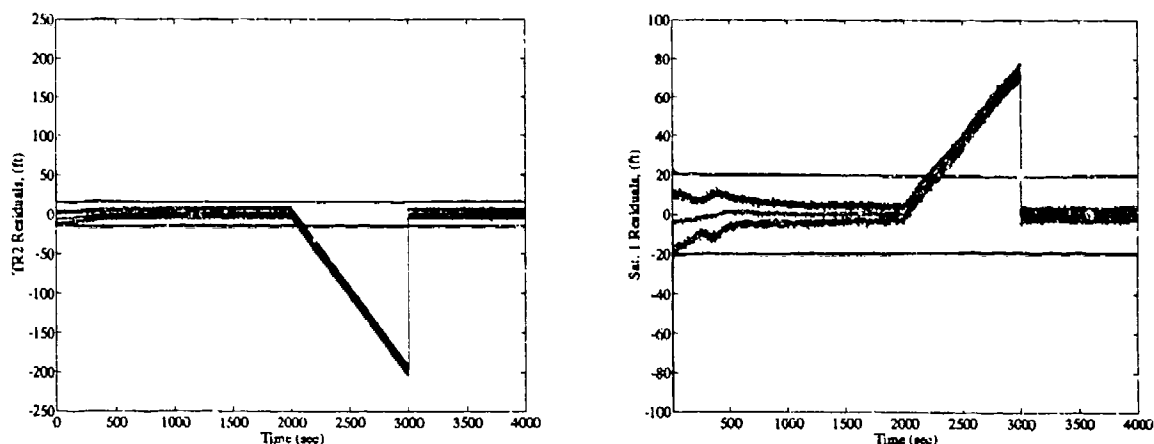


Figure 4.10 The Effect of a Transponder Ramp Failure on the Residuals of other Satellites and Transponder Measurements

One important aspect of any EKF is the state feedback into the \mathbf{F} and \mathbf{H} matrices of the linearized state and measurement equations. After each measurement update, the EKF will reevaluate the components of the \mathbf{F} and \mathbf{H} matrices based upon the new state estimate, $\hat{\mathbf{x}}(t_i^+)$, and newly declared trajectory segments emanating from that new estimate. This allows a failure added in one measurement to be reflected in the residuals of the other measurements. Figure 4.10 compares the residuals of a satellite and transponder when a second transponder signal has been failed. While the Satellite 1 measurement residual is biased by the Transponder 1 failure, the effect on the Transponder 2 measurement residual is more severe. Therefore, it is easier for a failed transponder measurement to affect other transponders rather than satellites. The translation of the failure into all the residuals increases the fidelity of the Chi-Square test and allows for consistent failure detection.

4.3.2.4 Transponder Noise Failure. Like the two other transponder failures, an increase in the transponder measurement noise is detected by the MNRS filter. Figures K.10, K.11, and K.12 show how the MNRS detects the failure in nine of the ten filters and switches to the tenth filter for navigation. The window size, $N = 3$, of the

Chi-Square test keeps the Chi-Square value above the threshold despite the occasional low value of failed measurement noise samples. Therefore, due to the size of the failure induced and the low-pass filtering effect of the Chi-Square window size, the MNRS FDIR algorithm recognizes and compensates for the noise failure over the entire period of the failure. Appendix G contains the number 1 transponder scalar residual plots for all ten NRS filters.

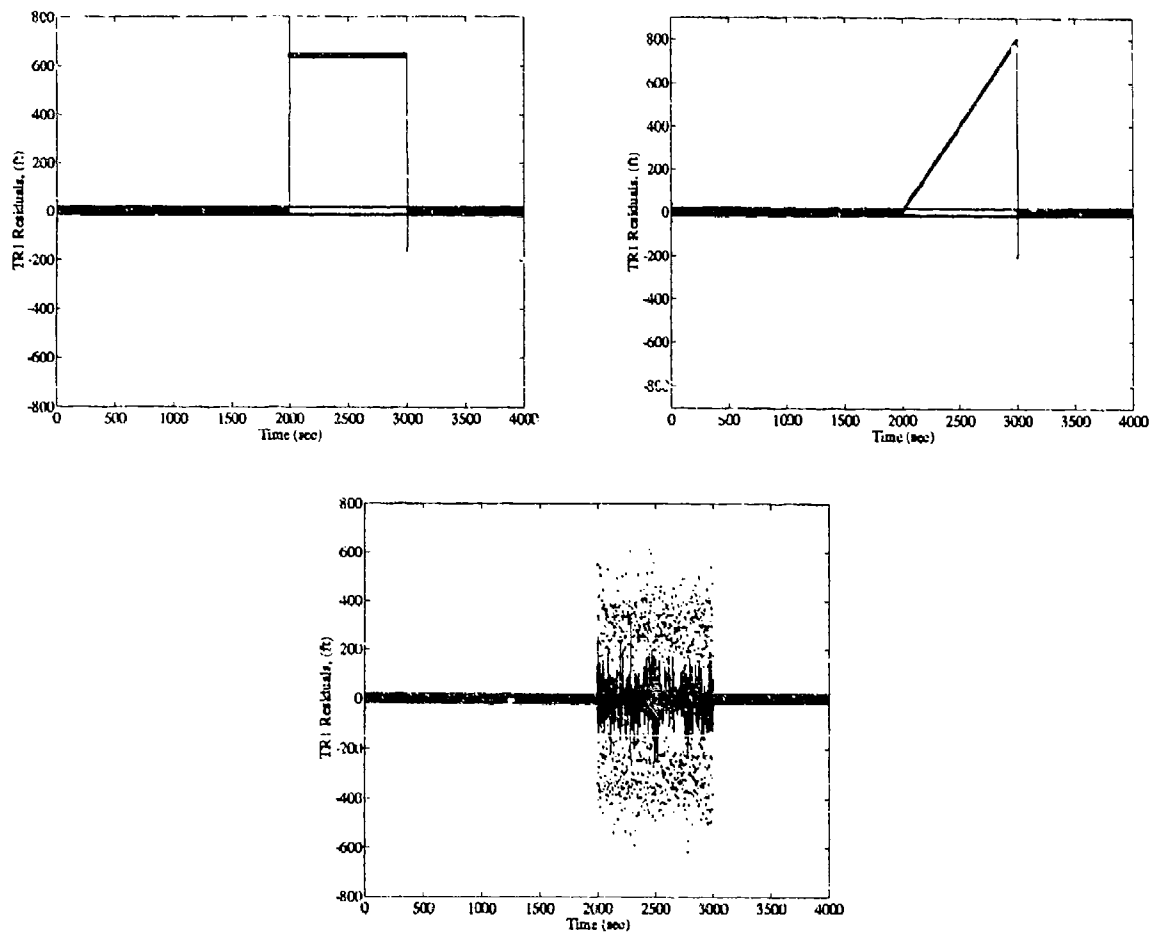


Figure 4.11 Comparison of Step, Ramp, and Noise Transponder Biases on the Scalar Residual of Transponder Measurement 1 of NRS filter 1

As with the previous two transponder measurement failures, the transponder noise failure generates a unique response in the scalar residual measurements. The failure type can be visually identified in the scalar residual of the failed measurement. Figure 4.11

compares the residuals of the Transponder 1 measurement for the transponder step, ramp, and noise failures. As can be seen, each of the failures causes a unique reaction in the scalar measurement residual. During the step bias, the Monte Carlo mean of the residual rises to a peak and then maintains a constant value through the failure period. During the ramp offset, the residual mean slowly slopes away from zero mean. Finally, the increase in the measurement noise increases the standard deviation of the residual without changing its mean value. The two matching filter techniques described in Sections 2.3.2 and 2.3.3 will attempt to exploit this behavior in the residuals to identify the type of failure affecting the system.

4.3.2.5 Satellite Bias Failure. While the MNRS FDIR algorithm correctly identifies and recovers from a satellite step bias failure, the residual results differ in significant ways from the transponder step bias results of Section 4.3.2.2. Before the differences are discussed, the Chi-Square test results for the satellite step bias failure run are presented. Figures K.13 and K.14 show the results for the individual Chi-Square tests conducted on each filter. As can be seen in the Chi-Square plots, NRS filter 3 is the only filter not receiving the failed measurement signal. Therefore, as seen in Figure K.15, the navigation correction switches to NRS filter 3 for the duration of the Chi-Square threshold violation. Therefore the MNRS FDIR algorithm does isolate, detect, and recover from a satellite step bias failure. Appendix H contains all transponder and satellite scalar residual plots for the ten NRS filters.

Despite the fact that the MNRS filter detects and recovers from both satellite and transponder step bias failures, the effect of the two failures on the scalar residuals differs drastically. As previously discussed in Section 4.3.2.3, the transponder step bias causes a continuous bias in the residual of the measurement residuals. This constant bias allows the Chi-Square test to detect the failure accurately over the entire failure lifetime. While this is true for a transponder failure, it is not true for a satellite failure. During the satellite step bias simulation, the magnitude of the Chi-Square test initially spikes when the failure first occurs. After the initial spike, the magnitude of the Chi-Square variable decreases back down to a failure-free level. Figure 4.12 shows how the filter actually "learns" the failure

and incorporates the erroneous data until it is impossible to distinguish that the failure is *still* occurring. This result is expected, since the GPS filter model contains error states that absorb the effect of the failure. The EKF interprets the satellite step bias failure as a change in the GPS user clock drift and bias error states. The filter uses these states to account for the failure effect, thereby removing the bias from the measurement residuals.

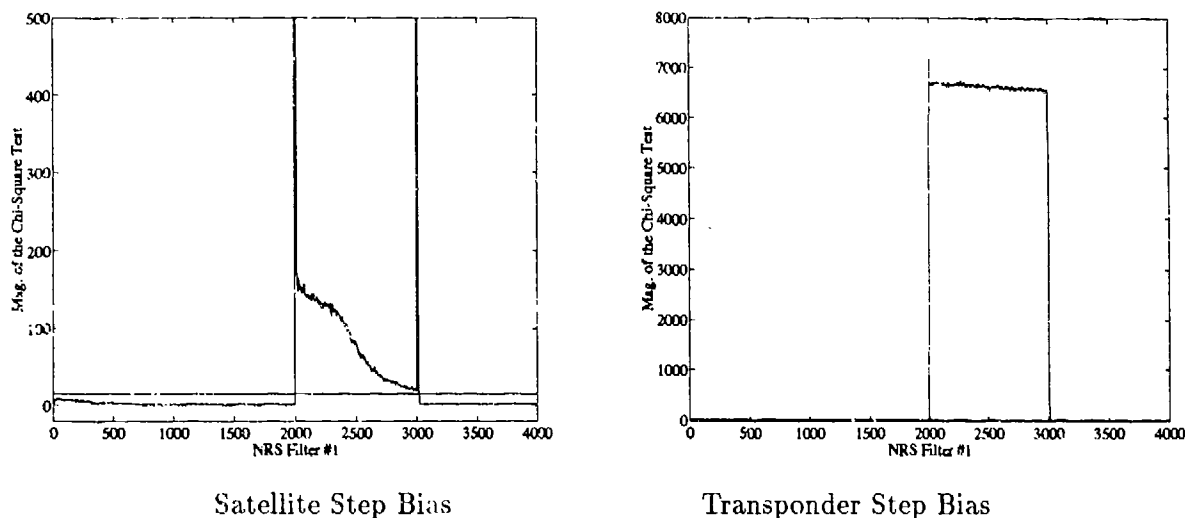


Figure 4.12 Effect of a Satellite Bias as Compared to a Transponder Bias on the Chi-Square Test of NRS Filter 1

While this effect is expected, it is not desirable. Had the failure duration been extended to 2000 seconds, the Chi-Square test would no longer have been a valid decision-maker between the different NRS filters. One by one each Chi-Square test would fall below threshold, despite the fact that the failure still existed. While this effect impairs the performance of the MNRS algorithm, a proposal is considered in Section 4.5 to use the results of a matching filter to compensate for the step bias failure. Given the implementation NRS filters used in this thesis, the MNRS algorithm cannot counteract the *learning* of the satellite step bias failure.

4.3.2.6 Satellite Ramp Failure. The results from the satellite ramp bias failure run are similar to the step offset results. While the MNRS FDIR algorithm is able to identify and recover from the failure correctly, the filter itself impedes the performance of the algorithm, compared to the transponder ramp bias results. The results achieved

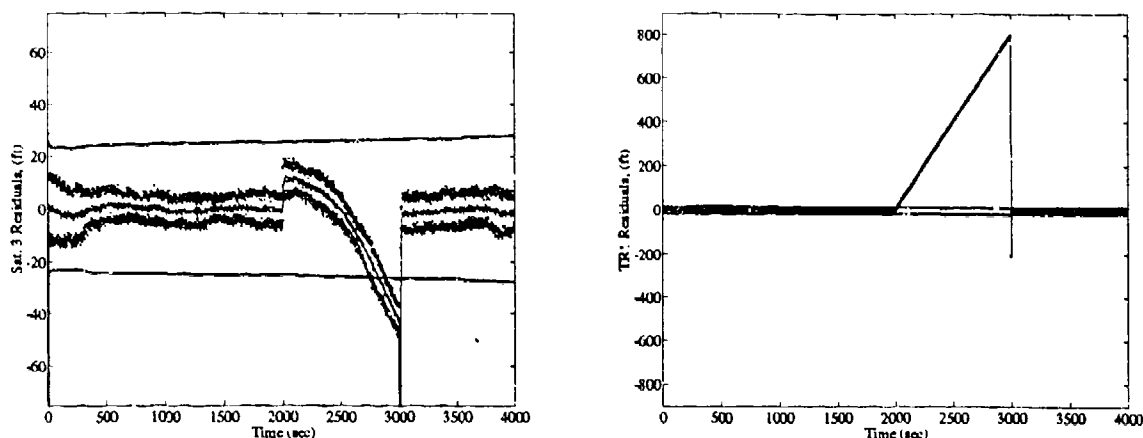


Figure 4.13 Comparison of Satellite and Transponder Residuals, Each Affected with a Additive Ramp Bias

by simulating the ten filters running in parallel can be seen in the Chi-Square plots in Figures K.16 and K.17. In these figures, each individual Chi-Square test experiences significant delay in detecting the failure. Figure 4.13 compares the scalar residuals of a satellite and transponder measurement, each biased with a ramp failure. While the transponder residual immediately slopes away from zero mean at a high rate, the satellite residual barely exceeds the filter-predicted standard deviation. Although the reader might suspect that this effect is caused by the relative precision of the two types of measurement updates, the relatively equal weighting of the tuning gains for satellite and transponder measurements does not support such a conclusion (see Table B.9). This subdued failure effect is directly related to the GPS user clock drift and bias states and their tendency to compensate for satellite failures. Despite the reduced response due to state compensation to failures, the MNRS can still detect and compensate for failures, thereby maintaining robust navigation (see Figure K.18). Appendix I contains satellite 3 scalar residual plots for the ten NRS filters.

4.3.2.7 Satellite Noise Failure. The final failure run in the thesis simulates an increase in measurement noise on a satellite measurement. While each of the NRS filters is able to maintain residual stability (zero mean), the increase in the variance of the measurement residuals is detected by the MNRS algorithm. As can be seen in Appendix J,

the Monte Carlo calculated variance of the residuals increases during the time of the failure. This increase in the variance of the residuals translates into large spikes in the output of the Chi-Square tests on each of the filters that receives the failed measurement. Figures K.19 and K.20 display the Chi-Square threshold tests for all ten filters. Figure K.21 shows the navigation switches to NRS filter 3 for the duration of the increase in measurement noise.

Each of the failure runs, the three transponder and three satellite runs demonstrate the effectiveness of the MNRS algorithm. By limiting the distribution of measurements to identical filters, failures are easily detected and recovered. There is no need for feedback since one of the NRS filters never updates with the failed measurement. Therefore the MNRS algorithm is successful in accomplishing failure detection, isolation, and recovery within the scope of the failure environment imposed in this research. This concludes the analysis of the MNRS FDIR algorithm.

4.4 Analysis of Measurement Geometry

This section analyzes the effects of satellite constellation geometry on the performance of individual NRS filters. Measurement geometry is defined as the positioning of the range measurement sources relative to the aircraft. Measurement geometry quite often dictates the quality of the navigation solution of an integrated navigation system. Due to the multiple model structure of the MNRS algorithm, the data collected in this thesis can also be used to analyze satellite geometry. This is possible since each filter in the MNRS model is updated with a unique set of measurements (see Table 1.1). In this analysis, the longitude error of an NRS filter is used to study the effect of satellite geometry on the performance of the state variables. Next the effect of satellite geometry on the residuals is also examined. This section demonstrates that satellite geometry impacts both state variable tracking and the FDIR performance of the NRS filter.

Of the two types of range measurements, GPS and RRS, GPS satellite geometry has a distinctly larger effect on the state variable tracking of the NRS model. To explore the impact of satellite geometry, the baseline simulation runs of NRS filters 1 and 4 are compared. Table 1.1 shows how these two filters differ by only one satellite measurement. The two filters are otherwise *identical* in all regards in this simulation study. Figure 4.14

shows the impact of the different satellite geometries on the longitude error of the two filters. The two plots in the figure display the difference between the Monte Carlo calculated mean ± 1 standard deviation of the difference between the filter longitude error-state and the truth model longitude error in a failure-free environment. The figure shows that

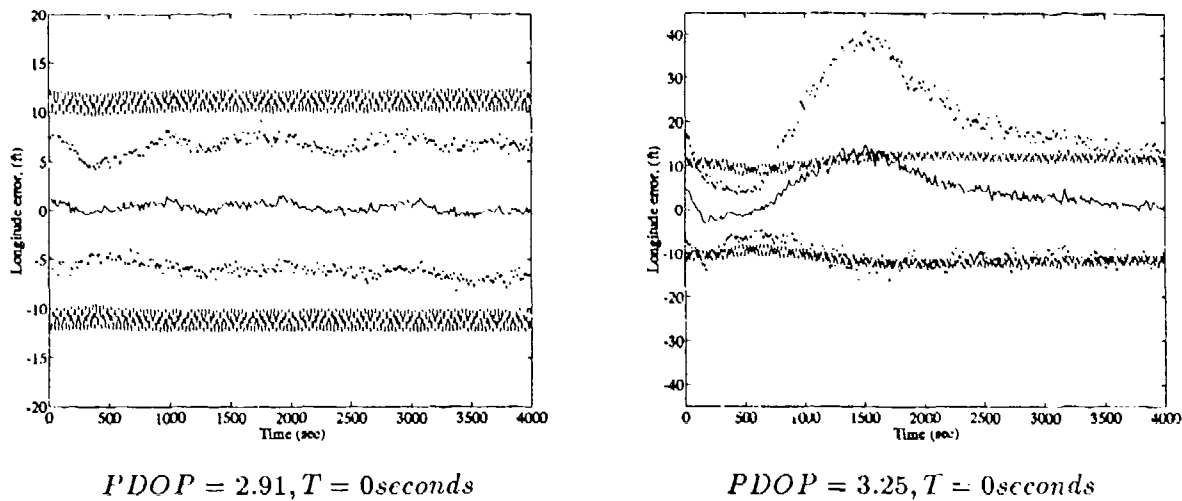


Figure 4.14 Comparison of Longitude Error States of NRS Filters with Different Satellite Geometry

NRS filter 4 does not maintain zero mean tracking of the truth model longitude error. The inability to track the error is caused by the relatively poor PDOP of NRS filter 4 compared to filter 1. The initial PDOP values for each of the two satellite measurement sets are recorded below the two plots in Figure 4.14. Since the direct relationship between PDOP and state estimation has been anticipated, the satellite measurement set with the best geometry is chosen to update the filter in the failure-free environment.

While poor satellite geometry has a negative effect on filter navigation, poor geometry actually makes failure detection easier. When a filter is affected with an unmodelled failure bias, filters with poor geometry reflect this failure more clearly in the residuals. On the other hand, when the geometry is good, the unmodelled failure bias is partially absorbed into the filter state estimates. Figure 4.15 compares a scalar residual of NRS filters 1 and 4, each affected with a satellite step failure. The filter with the poor geometry, filter 4, shows a larger failure bias in the scalar residual than filter 1. Therefore, when a Chi-Square

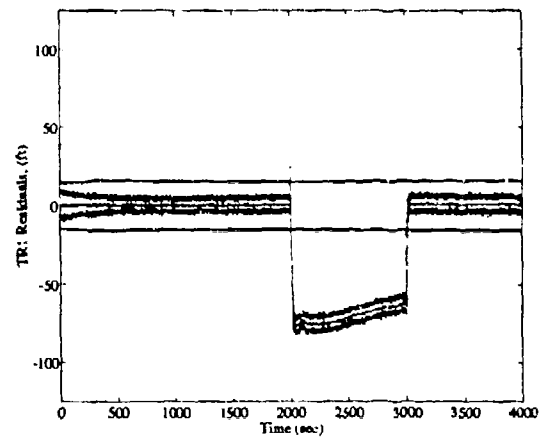
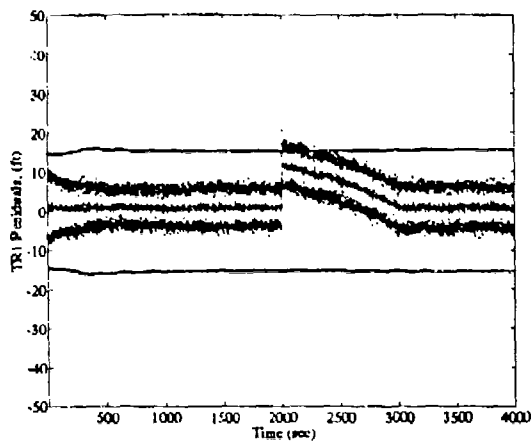


Figure 4.15 Comparison of Transponder 1 Scalar Measurement Residuals for NRS Filters 1 and 4 during a Step Bias Simulation Run

test is conducted on the two filters, the magnitude of the test will be greater in the filter with poor geometry. Figure 4.16 compares the two Chi-Square tests for filters 1 and 4 for the satellite step bias simulation. As can be seen in the figure, the Chi-Square test has an easier time detecting the failure in the filter with poor geometry. Therefore it is concluded that good measurement geometry hinders the ability of an FDIR algorithm to detect a failure.

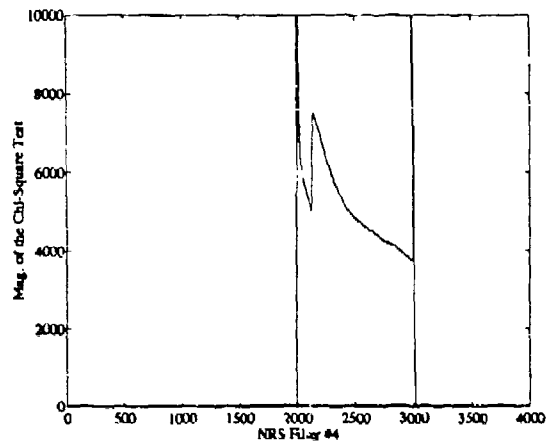
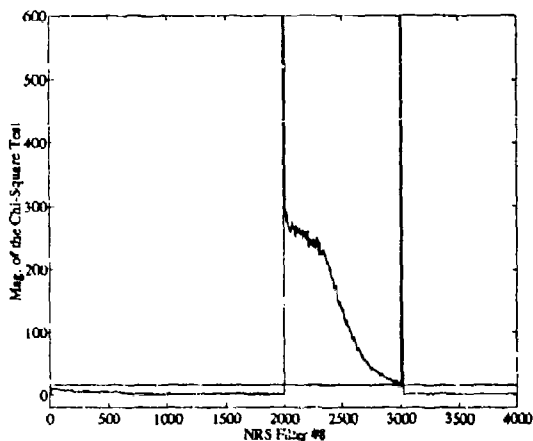


Figure 4.16 Comparison of Chi-Square Results for NRS Filters 1 and 4 during a Step Bias Simulation Run

The two conclusions about geometry reinforce preconceived notions about the NRS model. First, good geometry does improve the quality of state estimation. Since more accurate state performance is desirable, good geometry is also desirable. To go along with the improved state estimation, good geometry also helps the filter to track unmodelled failure biases. This tracking makes failure detection more difficult. While this section has addressed some of the issues involving measurement geometry considerations, the analysis is far from complete. Possibilities for future research endeavors will be discussed in Chapter V.

4.5 *Performance of the Matching Filter Algorithms*

The last research undertaken has been the work to develop an accurate matching filter algorithm to add to the MNRS FDIR algorithm. The purpose of the matching filter is to identify the type of measurement error on a known failed measurement. Within the scope of this thesis, it is desirable to identify between the three types of failures induced on the satellite and transponder range measurement signals: an additive step bias, an additive ramp offset, and an increase in measurement noise. Two matching filter algorithms are presented as methods for identifying the failure type: the generalized likelihood ratio (GLR) and the Chi-Square Pattern Recognition (CSPR) matching filters. The GLR technique implements a ratio of hypothesis probability densities to identify failures, while the CSPR attempts to match the shape of a failure offset to pre-determined failure model. Both these algorithms use the results of NRS filter 1 to attempt to identify the failure. The results of the GLR and then the CSPR matching filters are presented, followed by a comparison of the relative quality of the two algorithms. The goal of this analysis is to suggest the best direction for future FDIR work to pursue.

4.5.1 GLR Matching Filter Results. Overall, the results for the GLR algorithm have fallen short of expectations. While previous AFIT research has had limited success using the algorithm as a method to identify which measurement fails [19], the fidelity of the algorithm is not sufficient to identify the exact failure type. As stated in Section 2.3.2.2, the GLR matching filter relies on the magnitude of the $\Upsilon(t_i)$ function to determine the type

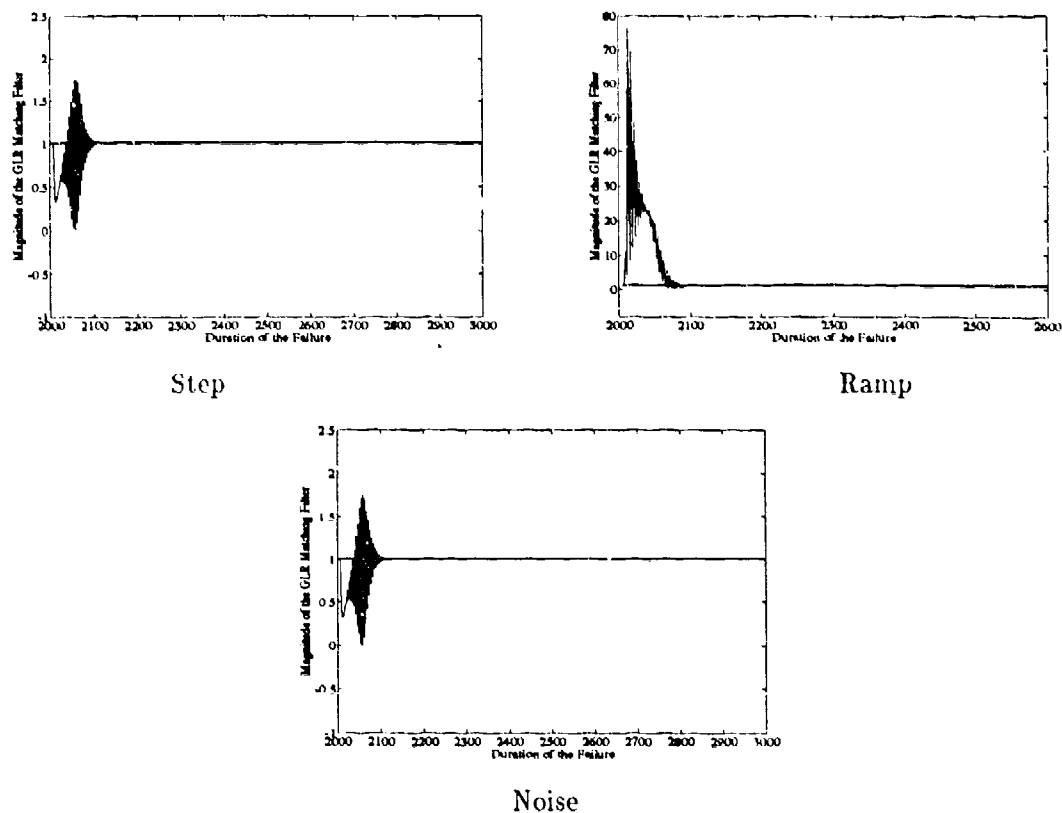


Figure 4.17 Comparison of GLR matching filter results, $\Upsilon(t_i)$, for a Transponder Step, Ramp, and Noise Failure

of failure. As explained in Equation (2.68), if the magnitude of $\Upsilon(t_i)$ is much greater than 1, the failure is a ramp offset. Similarly if the magnitude is much less than 1, the failure is identified as a step bias. If the magnitude remains in close vicinity of 1, neither step nor the ramp GLR failure model can identify the failure; therefore the algorithm assumes the failure is an increase in the measurement noise.

Despite the efforts to fine tune the algorithm, the GLR matching filter is unable to distinguish between different failure types. Figure 4.17 shows the response of the $\Upsilon(t_i)$ function for the three different transponder failures. While the algorithm seems to identify the ramp failure correctly, the step bias cannot be distinguished from an increase in measurement noise. In fact the GLR seems to believe both biases are increases in measurement noise. For the satellite measurement failures, all three failure types are identified as ramp

failures. As can be seen in Figure 4.18, the GLR is more confident in the step bias being a ramp failure than the actual ramp bias.

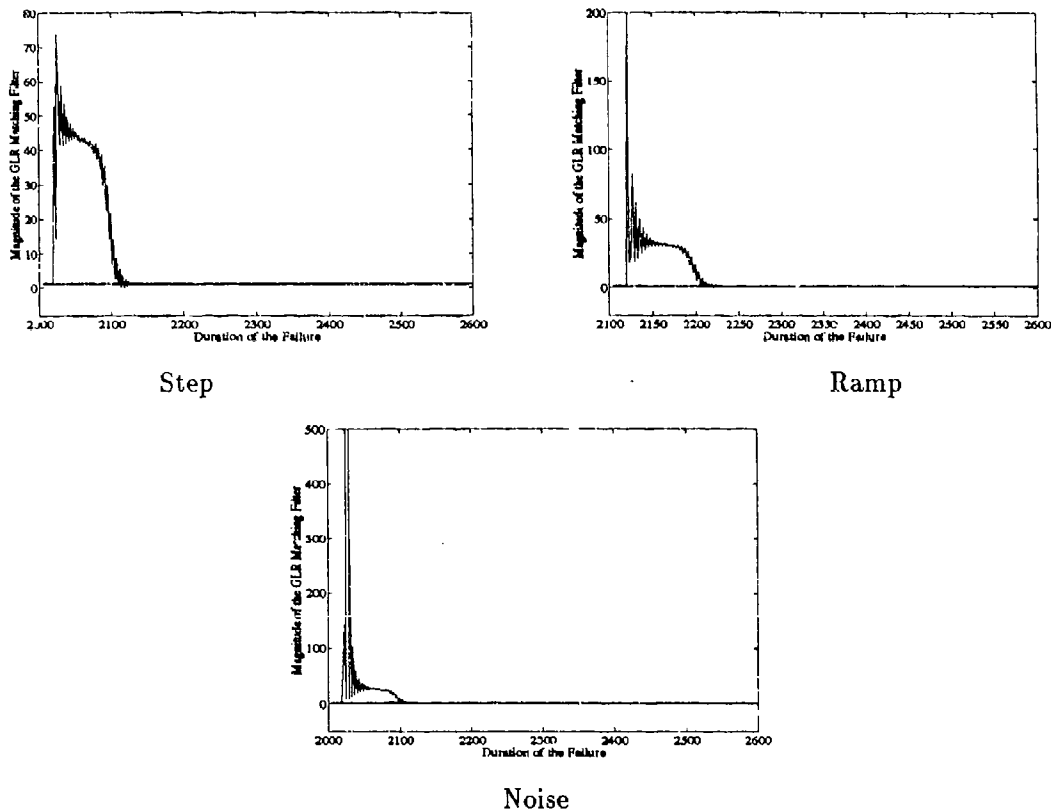


Figure 4.18 Comparison of GLR Matching Filter Results, $Y(t_i)$, for a Satellite Step, Ramp, and Noise Failure

While the results from the GLR simulation appear to be random, there is a common fault in the system that produces these results. The problem lies in a fundamental assumption, made in the derivation of the GLR equations. The GLR equations in Chapter II are derived for a discrete-time system model. Yet the model equations for the NRS filter are continuous-time state equations. This research assumes that the F matrix in the system dynamics equation can be discretized using a discretization algorithm described in Section 2.2.2. Unfortunately, this discretization process does not retain essential dynamics information necessary to match $g(t_i, \hat{\theta}_x)$ to the residuals in Equation (2.58). This discretization is hampered both by the coarse sampling time, 2 seconds, and the assumed first-order approximation (Section 2.2.2). This theoretical breakdown prevents

$Y(t_i)$ in Equation (2.67) from distinguishing between the ramp and step failures. Chapter V discusses possible solutions to this dilemma. Results for the GLR estimates of the failure magnitudes have not been included since the GLR matching filters cannot accurately specify failure type.

4.5.2 CSPR Matching Filter Results. While the results for the GLR algorithm are less than heartening, the CSPR matching filter provides hope for the future of failure type identification. While the GLR algorithm attempts to use information from the residuals, the \mathbf{F} matrix, and \mathbf{H} matrix to determine the failure type, the CSPR concentrates solely on the residual information to identify the failure. Therefore the CSPR algorithm is not limited by the validity of the discretization of the state dynamics matrix.

As explained in Section 3.4.4, the CSPR uses the Chi-Square output to determine the type of failure. The failure type is established by matching the shape of the failure portion of the Chi-Square output to the actual shape of the failure. Two criteria are used to evaluate the shape of the Chi-Square output. These criteria are the slope and the intercept of a first-order line fit of the Chi-Square output over the duration of the failure. To implement this criteria in real time, a new line fit is taken every time another Chi-Square output is above the failure threshold. Over time, the line fit will more accurately match the shape of the failure. By analyzing the slope and intercept of the line fits over time, the type of failure is determined.

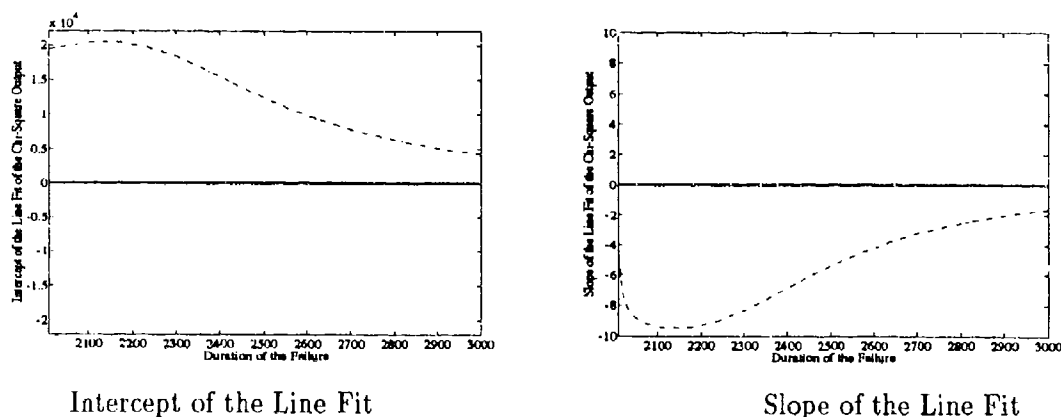


Figure 4.19 NRS Filter 1 Chi-Square Line Fit Data for a Satellite Step Bias

Table 4.4 Final Slope and Intercept Values for the Six Failure Runs

Type of Failure	Desired Slope	Final Slope	Desired Intercept	Final Intercept
Transponder Step Bias	Zero or Negative	0.397	Large and Positive	1.38×10^3
Transponder Ramp Bias	Positive	1.97	Zero or Negative	-4.22×10^3
Transponder Noise Increase	Zero Mean over Time	0.257×10^{-2}	Zero Mean over Time	1.68×10^2
Satellite Step Bias	Zero or Negative	-0.969	Large and Positive	2.76×10^3
Satellite Ramp Bias	Positive	1.22	Zero or Negative	-2.99×10^3
Satellite Noise Increase	Zero Mean over Time	0.380	Zero Mean over Time	5.49×10^2

To explain this somewhat confusing criterion, a satellite step bias simulation is examined. Figure 4.19 plots the slope and the intercept of the line fit to the Chi-Square output. During the early portion of the failure, the line fit has not established sufficient points of data, so the slope and intercept of the line fit deviate from expectations. However, as the failure continues, the slope and the intercept correspond to expectations described in Table 4.4. The intercept of the line fit is a large positive value and the slope of the line becomes slightly negative. This identifies the failure as a step bias on satellite measurement 1.

As explained earlier, the more data points available, the better the line fit matches the established criteria. Therefore, Table 4.4 documents the final slope and intercept values of the line fit for all the failure runs. The final slope and intercept are determined from the line fit incorporating all the Chi-Square outputs that are above the failure threshold. Therefore, there does exist significant delay in identification of failure type using this method. Appendix L contains the full slope and intercept plots for all the failure runs over the complete failure window.

While the CSPR can identify most failures, the algorithm is limited by its own definition. The CSPR will only correctly match the failure and the failure type when the Chi-Square output resembles the failure itself. While this is usually the case, there are exceptions, as can be seen in the Chi-Square output of NRS filter 1 during a satellite ramp

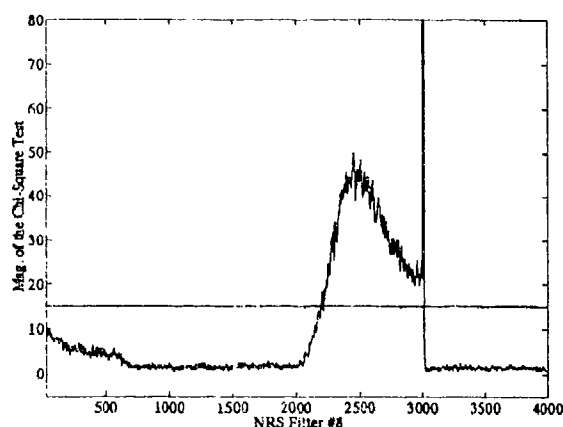


Figure 4.20 NRS Filter 1 Chi-Square Results for a Satellite Ramp Failure

failure. This Chi-Square output, Figure 4.20, does not resemble a slope over the time of failure. While the failure can be detected with the Chi-Square test, the CSPR matching algorithm will misinterpret the line fit, yielding an erroneous answer. Further study into the residual response to failures is necessary before a matching filter like the CSPR can accurately identify all failures.

While both the CSPR clearly outperforms the GLR matching filter, there are reasons to pursue both algorithms in future research. While the GLR requires a fundamental change in definition to be reapplied to the NRS filter model, pursuit of the algorithm does have a long-term payoff. Since the GLR equations are derived from the EKF model equations, the GLR has the potential to be applied to other models besides the NRS. On the other hand, the CSPR has provided reasonably successful identification using very little computational load and readily available inputs. However the CSPR algorithm has been developed from post-process analysis of one model affected with finite number of failure situations. If the scope of the research is broadened, there is no assurance that the CSPR matching filter will be able to perform consistently.

4.6 Chapter Summary

This ends the analysis of the data collected in the development of the MNRS FDIR algorithm. The following chapter summarizes the results of this thesis and presents recommendations for future research.

V. Conclusions and Recommendations

This chapter presents the conclusions reached in the research and development of the MNRS FDIR algorithm. The remarks will focus on the performance of the NRS single filter, MNRS FDIR algorithm, and matching filters. The chapter concludes with recommendations for future research endeavors.

5.1 NRS Filter Performance

The 15-State NRS filter has once again demonstrated accurate navigation corrections in the simulated environment. This NRS filter matches the performance of past filters, despite the reduction in the number transponder measurements from six to five and the correction of various errors in the NRS truth model. By keeping only the essential states with increased tuning gains, the filter can track the truth model with acceptable accuracy.

An examination of the model has shown that the GPS filter state variables are the key to this reduction in states. The coupling of the GPS drift and clock bias states with increased tuning gain will quickly compensate for unmodelled range biases. This is a highly beneficial quality when many of the truth model range biases are unmodelled in the filter. Yet this characteristic does hamper failure identification by tracking the occurrence of failures like any other unmodelled range bias. This is a trade-off consideration in developing the model. If the sensitivity of the GPS error states is reduced, the filter will not compensate for unmodelled range biases such as ionospheric and tropospheric delay. Since the choice in this thesis has always been to maintain the best possible performance in the failure-free environment, the FDIR technique must function in a less than optimal environment. This decision will be addressed again in Section 5.3.

The sensitivity of the model to changes in satellite geometry has also been investigated. The filter has shown that the PDOP of the satellites will affect the accuracy of the state estimation in the NRS implementation. Also, the negative effect of good PDOP has been demonstrated on the FDIR algorithm. The investigation of measurement geometry has been documented to encourage future research into this area.

5.2 MNRS FDIR Performance

The MNRS has shown that a simplified implementation of FDIR concepts can solve the problem of robust navigation in a failure environment. Researchers have a tendency to attempt to produce one solution for all problems (i.e. failure detection, isolation, and recovery). Both MMAE and the GLR algorithms have been proposed as cure-alls for the ailments of FDIR. Yet the MNRS algorithm exploits the positive qualities of several theories rather than just one, to attain robust navigation.

Throughout various research endeavors at AFIT, the Chi-Square test has shown an ability to detect the presence of a failure rapidly [2, 5]. Using residual information from all the measurements, the Chi-Square test detects failures as soon as the residuals deviate from their failure-free zero-mean behavior. The MNRS uses the Chi-Square variable only for its best quality, rapid and accurate detection of failures. The MNRS does not ask the Chi-Square test to isolate the failure or estimate the failure magnitude. This would be counterproductive, since the Chi-Square test does not perform these tasks well. Instead the MNRS uses other techniques better suited to isolation and recovery of the failure.

Failure isolation and recovery are accomplished with the use of multiple models. Assuming only single failures, the multiple models of the MNRS isolate the failed signal at the rate of Chi-Square detection. Rather than waiting for a more time-consuming and less reliable isolation algorithm, the MNRS has intermeshed the benefits of the Chi-Square test and multiple modelling to identify the failed measurement signal. Since the MNRS contains a model that does not receive updates from the failed measurement, recovery is as simple as switching the filter used to correct the INS solution.

The real limitations for the MNRS lie in the assumptions made in its development. The MNRS assumes that navigation with four satellites and five transponders is desirable. If more satellites and more transponders are used to update the filter, more filters are needed to maintain the FDIR capabilities of the algorithm. Also, reconfigurability must be considered after a failure has occurred. Once a measurement has failed, the entire MNRS should reset to be ready for another failure in the measurement signals. While these problems have reasonable solutions, to simulate the more realistic system is beyond the

scope this thesis. Nonetheless the performance of the MNRS demonstrates that exploiting the benefits of multiple FDIR techniques in one algorithm, can solve the problem of robust navigation in a single-failure environment.

5.3 Matching Filter Performance

The comparison of matching filter techniques has not yielded a clear favorite for future research. Neither algorithm demonstrates a clear superiority in performance. While the discrete-time GLR matching filter is able to detect failures, the continuous-to-discrete transformation of the \mathbf{F} dynamics matrix does not retain sufficient information to distinguish between failure types. Either the definitions of the failure models need to be altered or the GLR algorithm needs to be rederived using a continuous-time form of state equation. On the other hand, the CSPR matching filter does isolate failures accurately as long as the residuals retain the shape of the failure. For the transponder failures, the CSPR filter has absolutely no problems, because the Chi-Square output exactly matches the shape of the failure. This does not hold for the satellite failures. Due to the GPS filter state's tendency to absorb the bias induced in the failure and the ability of satellite geometry to affect the residual failure response, the Chi-Square test for satellite failures does not always match the shape of the failure itself. When Chi-Square output of a failure varies greatly from the expected shape of the failure, the CSPR will misinterpret the type of failure. This is a fundamental limitation of the CSPR matching filter.

Therefore the GLR still must be considered the more viable candidate for future failure type identification research. Algorithms like the CSPR are limited by the assumptions made in their implementation (i.e. the shape of the failure is independent of system dynamics). The GLR matching filter has the theoretical support to encourage future researchers to decipher the different problems encountered in this research. Overall failure type identification is still an extremely open ended area of research that needs further investigation.

5.4 Recommendations

To conclude this thesis, recommendations are made for future AFIT research. These ideas concentrate on improving the accuracy of the NRS model and furthering the development of failure type identification techniques.

- Verify the NRS filter in a new simulation environment. All past research on the NRS model has used consecutive generations of the same MSOFE Kalman filtering computer code. There is a definite need to transfer the code to a more interactive and modern programming environment. This process will allow reverification of past work and a more comprehensible base for future work.
- Analyze the impact of high dynamic profiles on the NRS model. CIGTF is interested in modelling the effects of GPS signal loss due to high dynamics maneuvers. By developing new models, the impact of high dynamics on the NRS model can be addressed.
- Continue the analysis of filter sensitivity to measurement geometry. This thesis only pursued this topic as an aside; measurement geometry is an important and unexplored area of research at AFIT.
- Implement a real-world NRS filter at AFIT. AFIT currently has the facilities and equipment to put together and test a hardware version of the NRS filter. Production of a real system would open a myriad of future research avenues for the NRS algorithm.
- Reverify the accuracy of the truth model for the NRS filter. The AFIT NRS truth model has remained unchanged for numerous years. The relative accuracy of the transponder measurements and the GPS measurements is of specific importance to filter performance.
- Research and develop a real velocity-aiding model for the NRS. Every generation of NRS research has shown that the filter goes unstable without velocity aiding. Yet an accurate velocity model has never been explored by AFIT. Therefore an imaginary model is still in place to maintain stability of the filter. The development of a velocity-aiding model will improve the validity of the NRS simulation results.

- Explore alternative failure identification algorithms. While the MNRS does provide nearly perfect FDIR performance, the multiple model structure carries a significant computational cost. Future research should explore the use of individual scalar residuals of a single filter to detect and isolate failures in place of a multiple model structure. The actual tuning of the single filter should be adjusted to enhance filter residual performance. By improving the performance of the residuals at the expense of the state variables, the filter now has the single purpose of FDI. This FDI filter can be run in parallel with a second EKF tuned for the most accurate state estimation. By feeding failure data from the FDI filter to the state estimation filter, a two-filter FDIR technique can be developed.
- Derive and implement a continuous-time GLR matching filter. A real problem in assessing the problems with the GLR matching filter is the discretization of the \mathbf{F} matrix inherent in the algorithm. This discretization may be causing the problems experienced with failure type identification. Therefore to implement the GLR equations properly, a continuous-time GLR matching filter is essential, rather than continuing to try to force feed a discrete time GLR matching filter into a sampled-data NRS EKF model.
- Pursue a new CSPR based algorithm to recognize patterns in the output of the EKF. Perhaps explore the possibilities of implementing a neural net to identify failure types.

Appendix A. Error State Definitions for the Truth and Filter Models

Tabular listings of the truth and filter models are presented. Tables A.1 and A.2 show the 39 INS states for the truth model, with the LN-93 state numbers given for reference to the Litton technical report on the INS [4]. Tables A.3 and A.4 list the RRS and GPS states respectively, and Table A.5 lists the states in the reduced-ordered NRS filter models.

Table A.1 39-State INS System Model: First 24 States

State Number	State Symbol	Definition	LN-93 State
1	$\delta\theta_x$	X-component of vector angle from true to computer frame	1
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame	2
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame	3
4	ϕ_x	X-component of vector angle from true to platform frame	4
5	ϕ_y	Y-component of vector angle from true to platform frame	5
6	ϕ_z	Z-component of vector angle from true to platform frame	6
7	δV_x	X-component of error in computed velocity	7
8	δV_y	Y-component of error in computed velocity	8
9	δV_z	Z-component of error in computed velocity	9
10	δh	Error in vehicle altitude above reference ellipsoid	10
11	δh_B	Total baro-altimeter correlated error	23
16	δh_L	Error in lagged inertial altitude	11
17	δS_3	Error in vertical channel aiding state	12
18	δS_4	Error in vertical channel aiding state	13
19	∇_{x_c}	X-component of accelerometer and velocity quantizer correlated noise	17
20	∇_{y_c}	Y-component of accelerometer and velocity quantizer correlated noise	18
21	∇_{z_c}	Z-component of accelerometer and velocity quantizer correlated noise	19
22	δg_x	X-component of gravity vector errors	20
23	δg_y	Y-component of gravity vector errors	21
24	δg_z	Z-component of gravity vector errors	22

Table A.2 39-State INS System Model: Second 19 States

State Number	State Symbol	Definition	LN-93 State
25	b_x	X-component of gyro drift rate repeatability	30
26	b_y	Y-component of gyro drift rate repeatability	31
27	b_z	Z-component of gyro drift rate repeatability	32
28	S_{gx}	X-component of gyro scale factor error	33
29	S_{gy}	Y-component of gyro scale factor error	34
30	S_{gz}	Z-component of gyro scale factor error	35
31	∇_{bx}	X-component of accelerometer bias repeatability	48
32	∇_{by}	Y-component of accelerometer bias repeatability	49
33	∇_{bz}	Z-component of accelerometer bias repeatability	50
34	S_{Ax}	X-component of accelerometer and velocity quantizer scale factor error	51
35	S_{Ay}	Y-component of accelerometer and velocity quantizer scale factor error	52
36	S_{Az}	Z-component of accelerometer and velocity quantizer scale factor error	53
37	S_{QA_x}	X-component of accelerometer and velocity quantizer scale factor asymmetry	54
38	S_{QA_y}	Y-component of accelerometer and velocity quantizer scale factor asymmetry	55
39	S_{QA_z}	Z-component of accelerometer and velocity quantizer scale factor asymmetry	56
40	μ_1	X accelerometer misalignment about Z-axis	66
41	μ_2	Y accelerometer misalignment about Z-axis	67
42	μ_3	Z accelerometer misalignment about Y-axis	68
43	σ_1	X-accelerometer misalignment about Y-axis	69

Table A.3 26-State RRS System Model

State Number	State Symbol	Definition
12	δR_b	Range error due to equipment bias
13	δv_b	Velocity error due to equipment bias
44	δP_{T1_x}	Transponder 1 x-component of position error
45	δP_{T1_y}	Transponder 1 y-component of position error
46	δP_{T1_z}	Transponder 1 z-component of position error
47	δR_{T1_a}	Transponder 1 range error due to atm propagation
48	δP_{T2_x}	Transponder 2 x-component of position error
49	δP_{T2_y}	Transponder 2 y-component of position error
50	δP_{T2_z}	Transponder 2 z-component of position error
51	δR_{T2_a}	Transponder 2 range error due to atm propagation
52	δP_{T3_x}	Transponder 3 x-component of position error
53	δP_{T3_y}	Transponder 3 y-component of position error
54	δP_{T3_z}	Transponder 3 z-component of position error
55	δR_{T3_a}	Transponder 3 range error due to atm propagation
56	δP_{T4_x}	Transponder 4 x-component of position error
57	δP_{T4_y}	Transponder 4 y-component of position error
58	δP_{T4_z}	Transponder 4 z-component of position error
59	δR_{T4_a}	Transponder 4 range error due to atm propagation
60	δP_{T5_x}	Transponder 5 x-component of position error
61	δP_{T5_y}	Transponder 5 y-component of position error
62	δP_{T5_z}	Transponder 5 z-component of position error
63	δR_{T5_a}	Transponder 5 range error due to atm propagation
64	δP_{T6_x}	Transponder 6 x-component of position error
65	δP_{T6_y}	Transponder 6 y-component of position error
66	δP_{T6_z}	Transponder 6 z-component of position error
67	δR_{T6_a}	Transponder 6 range error due to atm propagation

Table A.4 30-State GPS System Model

State Number	State Symbol	Definition
14	δR_{clk_u}	User clock bias
15	δD_{clk_u}	User clock drift
68	δR_{loop1}	SV 1 code loop error
69	δR_{trop1}	SV 1 tropospheric error
70	δR_{ion1}	SV 1 ionospheric error
71	$\delta R_{clk_{sv1}}$	SV 1 clock error
72	δx_{sv1}	SV 1 x-component of position error
73	δy_{sv1}	SV 1 y-component of position error
74	δz_{sv1}	SV 1 z-component of position error
75	δR_{loop2}	SV 2 code loop error
76	δR_{trop2}	SV 2 tropospheric error
77	δR_{ion2}	SV 2 ionospheric error
78	$\delta R_{clk_{sv2}}$	SV 2 clock error
79	δx_{sv2}	SV 2 x-component of position error
80	δy_{sv2}	SV 2 y-component of position error
81	δz_{sv2}	SV 2 z-component of position error
82	δR_{loop3}	SV 3 code loop error
83	δR_{trop3}	SV 3 tropospheric error
84	δR_{ion3}	SV 3 ionospheric error
85	$\delta R_{clk_{sv3}}$	SV 3 clock error
86	δx_{sv3}	SV 3 x-component of position error
87	δy_{sv3}	SV 3 y-component of position error
88	δz_{sv3}	SV 3 z-component of position error
89	δR_{loop4}	SV 4 code loop error
90	δR_{trop4}	SV 4 tropospheric error
91	δR_{ion4}	SV 4 ionospheric error
92	$\delta R_{clk_{sv4}}$	SV 4 clock error
93	δx_{sv4}	SV 4 x-component of position error
94	δy_{sv4}	SV 4 y-component of position error
95	δz_{sv4}	SV 4 z-component of position error

Table A.5 15-State Reduced-Order Filter Model

State Number	State Symbol	Definition
1	$\delta\theta_x$	X-component of vector angle from true to computer frame
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame
4	ϕ_x	X-component of vector angle from true to platform frame
5	ϕ_y	Y-component of vector angle from true to platform frame
6	ϕ_z	Z-component of vector angle from true to platform frame
7	δV_x	X-component of error in computed velocity
8	δV_y	Y-component of error in computed velocity
9	δV_z	Z-component of error in computed velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_B	Total baro-altimeter correlated error
12	δR_b	Range error due to equipment bias
13	δv_b	Velocity error due to equipment bias
14	δx_{clk_b}	User clock bias
15	$\delta x_{clk_{dr}}$	User clock drift

Appendix B. Dynamics Matrices and Noise Values

B.1 Definition of Dynamics Matrices

In Chapter 3, the truth and filter model dynamics matrices are defined by the submatrices, F_{Filter} , $F_{INS_{i1}}$, $F_{INS_{i2}}$, F_{RRS_i} , and F_{GPS_i} of Equation (3.3). The F_{Filter} represents the filter dynamics matrix, which is also a submatrix in the larger truth model dynamics matrix. The other three matrices represent the additional truth model non-zero portions of the F matrix that simulate the real world. Tables B.1, B.2, B.3, B.4, and B.5 contain the non-zero elements of the dynamics submatrices F_{Filter} , $F_{INS_{i1}}$, $F_{INS_{i2}}$, F_{RRS_i} , and F_{GPS_i} respectively. All the undeclared variables shown in the following tables are defined in the LN-93 technical report, along with their units [4]. The structure of the dynamics matrices below correspond to the truth model state definitions in Appendix A.

Table B.1 Elements of the Dynamics Submatrix F_{Filter}

Element	Term	Element	Term
(1,3)	$-\rho_y$	(1,8)	$-C_{RY}$
(2,3)	ρ_x	(2,7)	C_{RX}
(3,1)	ρ_y	(3,2)	$-\rho_x$
(4,2)	$-\Omega_z$	(4,3)	Ω_y
(4,5)	ω_{it_z}	(4,6)	$-\omega_{it_y}$
(4,8)	$-C_{RY}$	(5,1)	Ω_z
(5,3)	$-\Omega_x$	(5,4)	$-\omega_{it_x}$
(5,6)	ω_{it_z}	(5,7)	C_{RX}
(6,1)	$-\Omega_y$	(6,2)	Ω_x
(6,4)	ω_{it_y}	(6,5)	$-\omega_{it_x}$
(7,1)	$-2V_y\Omega_y - 2V_z\Omega_z$	(7,2)	$2V_y\Omega_x$
(7,3)	$2V_z\Omega_y$	(7,5)	$-A_z$
(7,6)	A_y	(7,7)	$-V_zC_{RX}$
(7,8)	$2\Omega_z$	(7,9)	$-\rho_y - 2\Omega_y$
(8,1)	$2V_x\Omega_y$	(8,2)	$-2V_x\Omega_x - 2V_z\Omega_z$
(8,3)	$2v_z\Omega_y$	(8,4)	A_z
(8,6)	$-A_x$	(8,7)	$-2\Omega_z$
(8,8)	$-V_zC_{RY}$	(8,9)	$\rho_x + 2\Omega_x$
(9,1)	$2V_x\Omega_z$	(9,2)	$2V_y\Omega_z$
(9,3)	$-2V_y\Omega_y - 2V_x\Omega_x$	(9,4)	$-A_y$
(9,5)	A_x	(9,7)	$\rho_y + 2\Omega_y + V_zC_{RX}$
(9,8)	$-\rho_x - 2\Omega_x + V_yC_{RY}$	(9,10)	$2g_o/a$
(11,11)	$-\beta_{\delta h_c}$	(14,15)	$1 \text{ ft}^2/sec$

Table B.2 Elements of the Dynamics Submatrix F_{INSII}

Element	Term	Element	Term	Element	Term
(9,16)	$-k_2$	(9,17)	-1	(9,18)	k_2
(10,9)	1	(10,16)	$-k_1$	(10,18)	$k_1 - 1$
(16,10)	1	(16,16)	-1	(17,16)	k_3
(17,18)	$-k_3$	(18,10)	k_4	(18,16)	$-k_4$
(7,19)	C_{11}	(7,20)	C_{12}	(7,21)	C_{13}
(7,22)	1	(8,19)	C_{21}	(8,20)	C_{22}
(8,21)	C_{23}	(8,23)	1	(9,19)	C_{31}
(9,20)	C_{32}	(9,21)	C_{33}	(9,24)	1
(9,11)	k_2	(10,11)	k_1	(17,11)	$-k_3$
(18,11)	$k_4/600$	(18,18)	$k_4 - 1$	(9,43)	$C_{33}A_x^B$
(4,25)	C_{11}	(4,26)	C_{12}	(4,27)	C_{13}
(4,28)	$C_{11}\omega_{ib_x}$	(4,29)	$C_{12}\omega_{ib_y}$	(4,30)	$C_{13}\omega_{ib_z}$
(5,25)	C_{21}	(5,26)	C_{22}	(5,27)	C_{23}
(5,28)	$C_{21}\omega_{ib_x}$	(5,29)	$C_{22}\omega_{ib_y}$	(5,30)	$C_{23}\omega_{ib_z}$
(6,25)	C_{31}	(6,26)	C_{32}	(6,27)	C_{33}
(6,28)	$C_{31}\omega_{ib_x}$	(6,29)	$C_{32}\omega_{ib_y}$	(6,30)	$C_{33}\omega_{ib_z}$
(7,31)	C_{11}	(7,32)	C_{12}	(7,33)	C_{13}
(7,34)	$C_{11}A_x^B$	(7,35)	$C_{12}A_y^B$	(7,36)	$C_{13}A_z^{B'}$
(7,37)	$C_{11} A_x^B $	(7,38)	$C_{12} A_y^B $	(7,39)	$C_{13} A_z^{B'} $
(7,40)	$C_{11}A_y^B$	(7,41)	$-C_{12}A_x^B$	(7,42)	$C_{13}A_y^B$
(7,43)	$C_{13}A_x^B$	(8,31)	C_{21}	(8,32)	C_{22}
(8,33)	C_{23}	(8,34)	$C_{21}A_x^B$	(8,35)	$C_{22}A_y^B$
(8,36)	$C_{23}A_z^{B'}$	(8,37)	$C_{21} A_x^B $	(8,38)	$C_{22} A_y^B $
(8,39)	$C_{23} A_z^{B'} $	(8,40)	$C_{21}A_y^B$	(8,41)	$-C_{22}A_x^B$
(8,42)	$C_{23}A_y^B$	(8,43)	$C_{23}A_x^B$	(9,31)	C_{31}
(9,32)	C_{32}	(9,33)	C_{33}	(9,34)	$C_{31}A_x^B$
(9,35)	$C_{32}A_y^B$	(9,36)	$C_{33}A_z^{B'}$	(9,37)	$C_{31} A_x^B $
(9,38)	$C_{32} A_y^B $	(9,39)	$C_{33} A_z^{B'} $	(9,40)	$C_{31}A_x^B$
(9,41)	$-C_{32}A_z^B$	(9,42)	$C_{33}A_y^B$		

Table B.3 Elements of the Dynamics Submatrix F_{INS_t}

Element	Term	Element	Term	Element	Term
(19,19)	$-\beta_{\nabla_{x_c}}$	(20,20)	$-\beta_{\nabla_{y_c}}$	(21,21)	$-\beta_{\nabla_{z_c}}$
(22,22)	$-\beta_{\delta g_x}$	(23,23)	$-\beta_{\delta g_y}$	(24,24)	$-\beta_{\delta g_z}$

Table B.4 Elements of the Dynamics Submatrix F_{RRS}

Element	Term	Element	Term
(47,47)	$-1/300 \text{ ft}^2/\text{sec}$	(51,51)	$-1/300 \text{ ft}^2/\text{sec}$
(55,55)	$-1/300 \text{ ft}^2/\text{sec}$	(59,59)	$-1/300 \text{ ft}^2/\text{sec}$
(63,63)	$-1/300 \text{ ft}^2/\text{sec}$	(67,67)	$-1/300 \text{ ft}^2/\text{sec}$

Table B.5 Elements of the Dynamics Submatrix F_{GPS}

Element	Term	Element	Term	Element	Term
(68,68)	$-1 \text{ ft}^2/\text{sec}$	(69,69)	$-1/500 \text{ ft}^2/\text{sec}$	(70,70)	$-1/1500 \text{ ft}^2/\text{sec}$
(73,75)	$-1 \text{ ft}^2/\text{sec}$	(76,76)	$-1/500 \text{ ft}^2/\text{sec}$	(77,77)	$-1/1500 \text{ ft}^2/\text{sec}$
(82,82)	$-1 \text{ ft}^2/\text{sec}$	(83,83)	$-1/500 \text{ ft}^2/\text{sec}$	(84,84)	$-1/1500 \text{ ft}^2/\text{sec}$
(89,89)	$-1 \text{ ft}^2/\text{sec}$	(90,90)	$-1/500 \text{ ft}^2/\text{sec}$	(91,91)	$-1/1500 \text{ ft}^2/\text{sec}$

B.2 Elements of the Process Noise and Measurement Noise Matrices

This section defines the dynamics noise strengths and measurement noise variances for the truth and filter models. The truth model non-zero dynamics noise strength are defined in Tables B.6 and B.7. These noise strengths correspond to the driving noises w_{Filter_t} , w_{INS_t} , w_{RRS_t} , and w_{GPS_t} in Equation (3.3). Note that the σ^2 terms in Table B.6 are variable names as defined in the Litton technical report and do not represent variance terms typically associated with σ^2 . The filter dynamics driving noise terms implemented after filter tuning are listed in Table B.8. Finally, the measurement noise variances used in the truth and filter models are presented in Table B.9.

Table B.6 Elements of Truth Model Process Noise Submatrix for the INS Truth Model

Element	Term	Element	Term		
(4,4)	$\sigma_{\eta_{b_x}}^2$	(5,5)	$\sigma_{\eta_{b_y}}^2$	(6,6)	$\sigma_{\eta_{b_z}}^2$
(7,7)	$\sigma_{\eta_{A_x}}^2$	(8,8)	$\sigma_{\eta_{A_y}}^2$	(9,9)	$\sigma_{\eta_{A_z}}^2$
(11,11)	$2\beta_{\delta h_c} \sigma_{\delta h_c}^2$	(19,19)	$2\beta_{\nabla_{x_c}} \sigma_{\nabla_{x_c}}^2$	(20,20)	$2\beta_{\nabla_{y_c}} \sigma_{\nabla_{y_c}}^2$
(21,21)	$2\beta_{\nabla_{x_c}} \sigma_{\nabla_{x_c}}^2$	(22,22)	$2\beta_{\delta g_x} \sigma_{\delta g_x}^2$	(23,23)	$2\beta_{\delta g_y} \sigma_{\delta g_y}^2$
(24,24)	$2\beta_{\delta g_x} \sigma_{\delta g_x}^2$				

Table B.7 Elements of Truth Model Process Noise for RRS and GPS States

Element	Term	Element	Term		
(47,47)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$	(51,51)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$	(55,55)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$
(59,59)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$	(63,63)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$	(67,67)	$6.667 \times 10^{-13} \text{ ft}^2/\text{sec}$
(68,68)	$0.5 \text{ ft}^2/\text{sec}$	(69,69)	$0.004 \text{ ft}^2/\text{sec}$	(70,70)	$0.004 \text{ ft}^2/\text{sec}$
(75,75)	$0.5 \text{ ft}^2/\text{sec}$	(76,76)	$0.004 \text{ ft}^2/\text{sec}$	(77,77)	$0.004 \text{ ft}^2/\text{sec}$
(82,82)	$0.5 \text{ ft}^2/\text{sec}$	(83,83)	$0.004 \text{ ft}^2/\text{sec}$	(84,84)	$0.004 \text{ ft}^2/\text{sec}$
(89,89)	$0.5 \text{ ft}^2/\text{sec}$	(90,90)	$0.004 \text{ ft}^2/\text{sec}$	(91,91)	$0.004 \text{ ft}^2/\text{sec}$

Table B.8 Filter Process Noise Q Values

Element	Term	Element	Term
(1,1)	$1.2 \times 10^{-13} \text{ rad}^2/\text{sec}$	(2,2)	$1.5 \times 10^{-13} \text{ rad}^2/\text{sec}$
(3,3)	$0.0 \text{ rad}^2/\text{sec}$	(4,4)	$5 \text{ rad}^2/\text{sec}$
(5,5)	$5 \text{ rad}^2/\text{sec}$	(6,6)	$85 \text{ rad}^2/\text{sec}$
(7,7)	$15000 \text{ ft}^2/\text{sec}^3$	(8,8)	$15000 \text{ ft}^2/\text{sec}^3$
(9,9)	$110000 \text{ ft}^2/\text{sec}^3$	(10,10)	$300 \text{ ft}^2/\text{sec}^2$
(11,11)	$300 \text{ ft}^2/\text{sec}^2$	(12,12)	$50 \text{ ft}^2/\text{sec}^2$
(13,13)	$0.0 \text{ ft}^2/\text{sec}^3$	(14,14)	$70 \text{ \& } 40 \text{ ft}^2/\text{sec}^2$
(15,15)	$0.5 \times 10^{-15} \text{ ft}^2/\text{sec}^2$		

Table B.9 Truth and Filter Measurement Noises R Values

Measurement	Truth Noise	Filter Noise
Baro Altimeter	2500 ft^2	3500 ft^2
Doppler	$0.02 \text{ ft}^2/\text{sec}^2$	$0.02 \text{ ft}^2/\text{sec}^2$
Transponders	4 ft^2	100 ft^2
Satellites	2 ft^2	75 ft^2

Appendix C. Tuning Plots for an NRS Filter

These plots are the baseline filter tuning plots for a single NRS filter. The plots demonstrate the effect of the tuning values implemented in all ten of the filters used in the MNRS multiple model. These tuning plots have been run using satellites 1-4 and transponders 1-5.

The plots in this section are error mean ± 1 standard deviation plots. They reveal the quality of the filter tuning and the filter's ability to track the truth model. It should be noted that the plots in this appendix are not comparing the actual filter and truth states. The plots depict the filter and truth states translated into the more commonly recognized navigation variables (latitude, longitude, north velocity, altitude, etc, etc). The translation of the variables has been used before and is documented in past AFIT theses [14, 18].

The solid center line on the following plots is the *mean error* time history for the applicable navigation variable. The *mean error* is defined as the Monte Carlo average of the difference between the filter estimate of a state and the actual truth state. The following equation is the mathematical definition of the *mean error* [9, 19, 13]:

$$\hat{M}_e(t_i) = \frac{1}{N} \sum_{j=1}^N e_j(t_i) = \frac{1}{N} \sum_{j=1}^N \{\hat{x}_j(t_i) - x_{true,j}(t_i)\} \quad (C.1)$$

where $\hat{x}_j(t_i)$ is the filter-computed estimate of a given navigation variable and $x_{true,j}(t_i)$ is the *truth* model value of the same variable, at time t_i , for run j . N is the number of Monte Carlo runs in the simulation (15 in this thesis).

In addition to the center trace, two more *pairs* of lines are plotted. The first pair is the *Mean \pm Sigma* (represented by the \cdots lines in the plots). These lines are symmetrically displaced about the *mean error*, $\hat{M}_e(t_i)$. The *Mean \pm Sigma* is the sum of the previously defined mean, $\hat{M}_e(t_i)$, and the actual filter standard deviation $\pm\sqrt{P_e(t_i)}$, where $P_e(t_i)$ is the *true* error variance at time t_i . The true standard deviation is calculated from the following equation [9, 19]:

$$\sigma_{true}(t_i) = \sqrt{P_e(t_i)} = \sqrt{\frac{1}{N-1} \sum_{j=1}^N e_j^2(t_i) - \frac{N}{N-1} \hat{M}_e^2(t_i)} \quad (C.2)$$

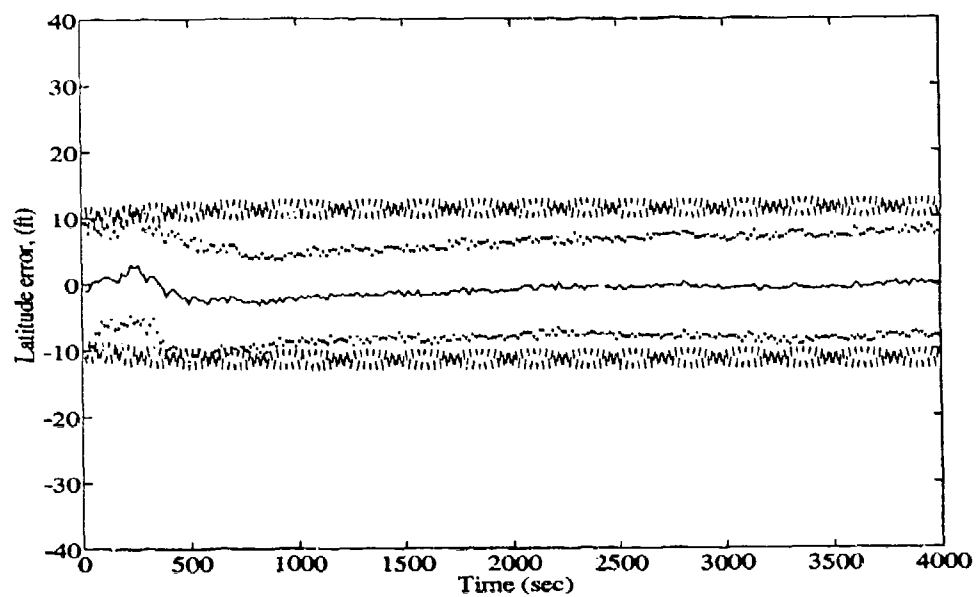
N is the number of runs in the Monte Carlo simulation, and $\hat{M}_e^2(t_i)$ is the square of the mean of a given state at each time of interest.

The second pair of traces (— — lines) represent the filter-computed $\pm \sigma_{filter}$ values for the same states and are symmetrically displaced about zero because the filter “believes” that it is producing zero-mean errors [10, 19]. These quantities are propagated and updated in the MSOFE [13, 18] software, using the covariance propagation equation shown in Chapter II. These plotted lines represent the filter’s estimate of its own error.

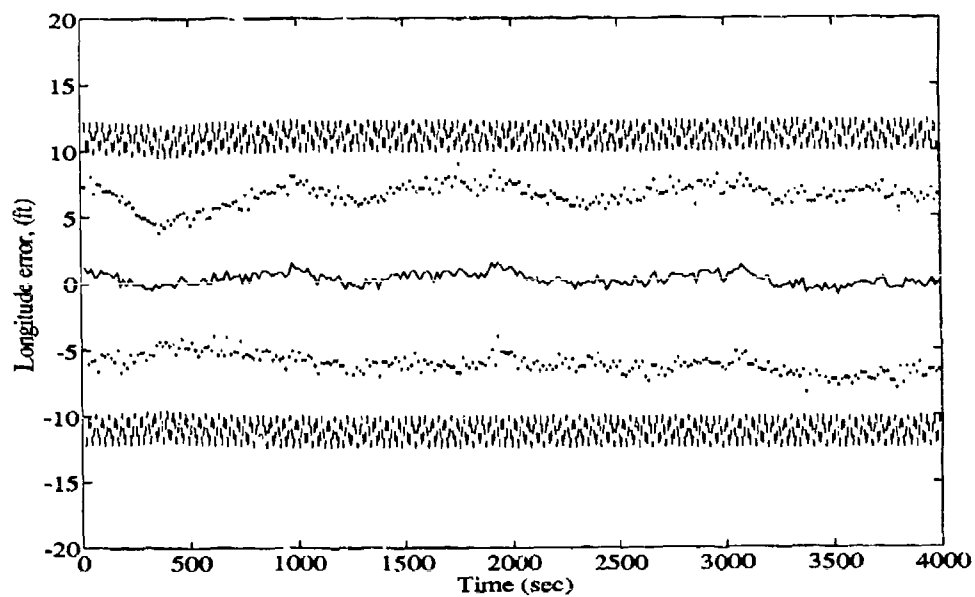
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. This section contains error state covariance plots for fourteen navigation variables of interest.

Table C.1 Legend for Filter Tuning Plots

Symbol	Definition
— Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
— — Dashed Line	\pm Filter-Predicted Sigma

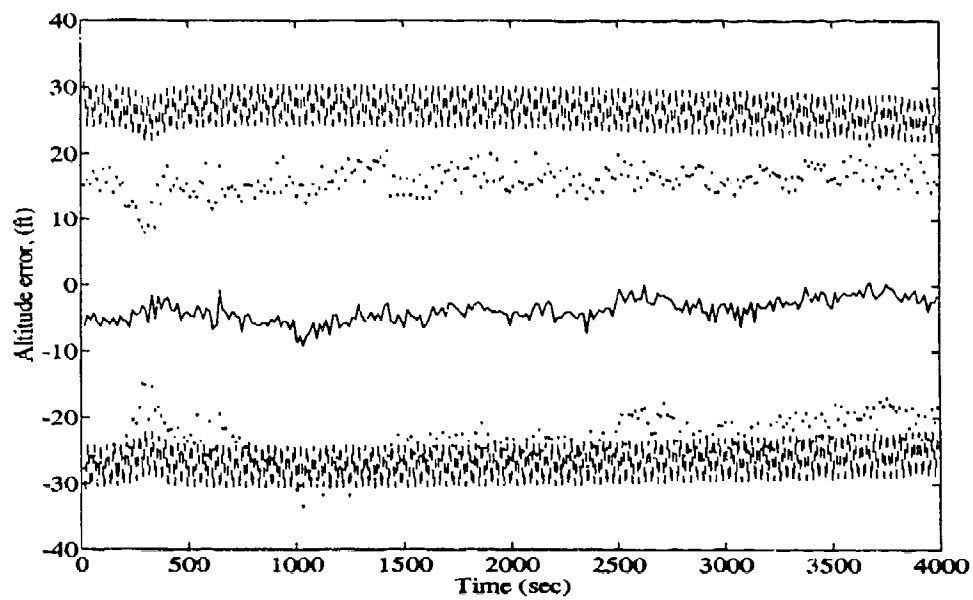


(a)

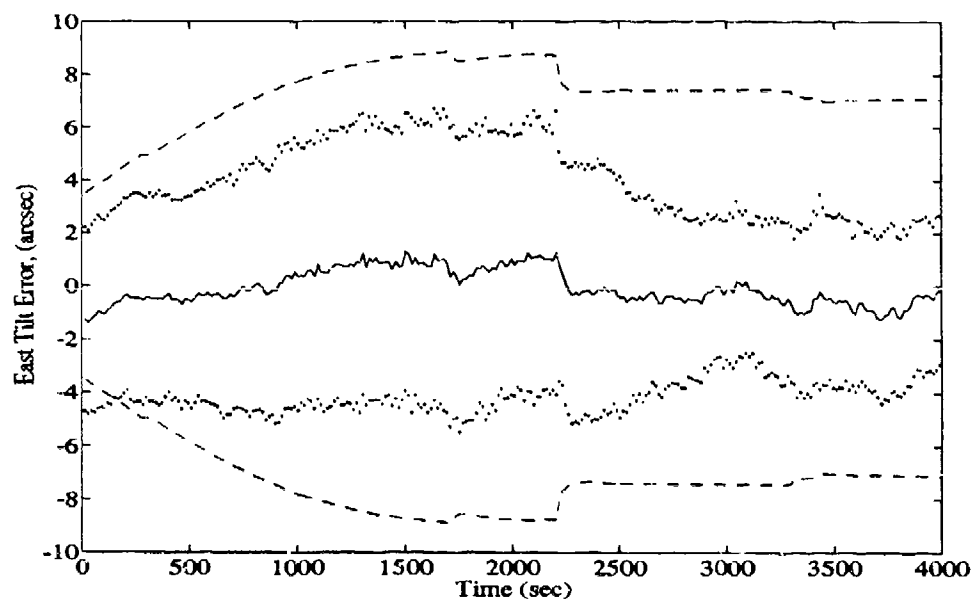


(b)

Figure C.1 Latitude and Longitude Error Filter tuning Plots

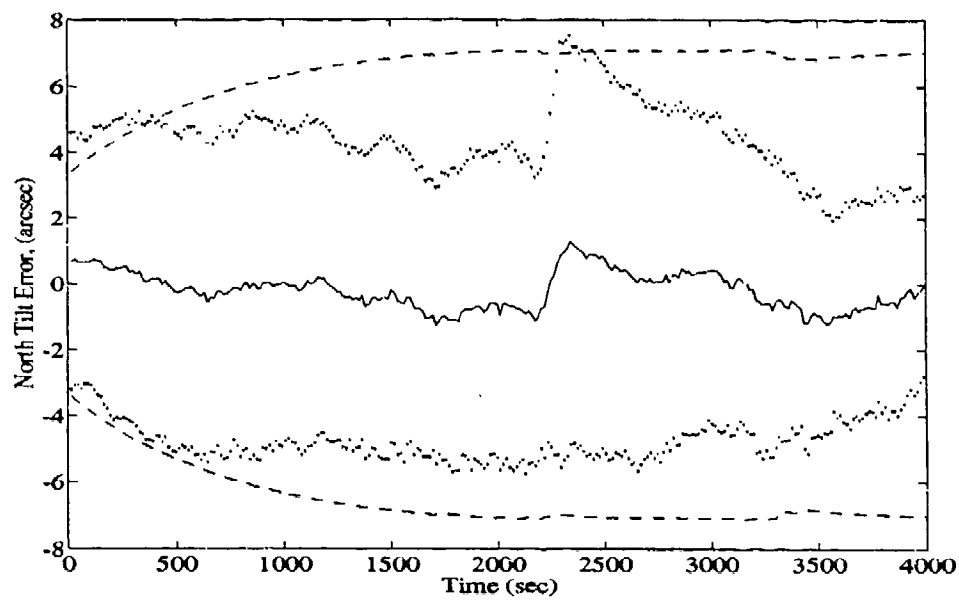


(a)

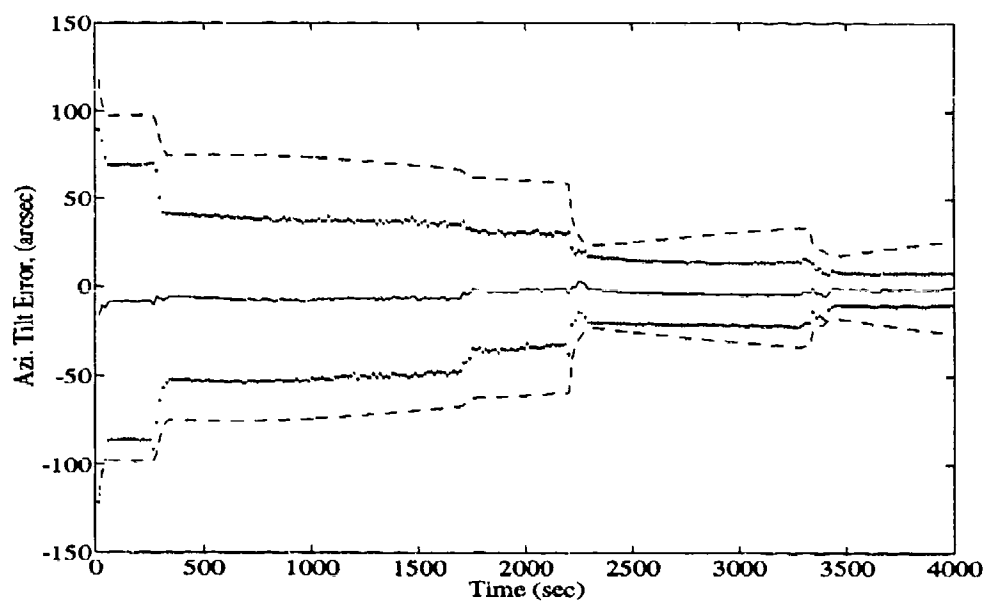


(b)

Figure C.2 Altitude and North Tilt Error Filter Tuning Plots

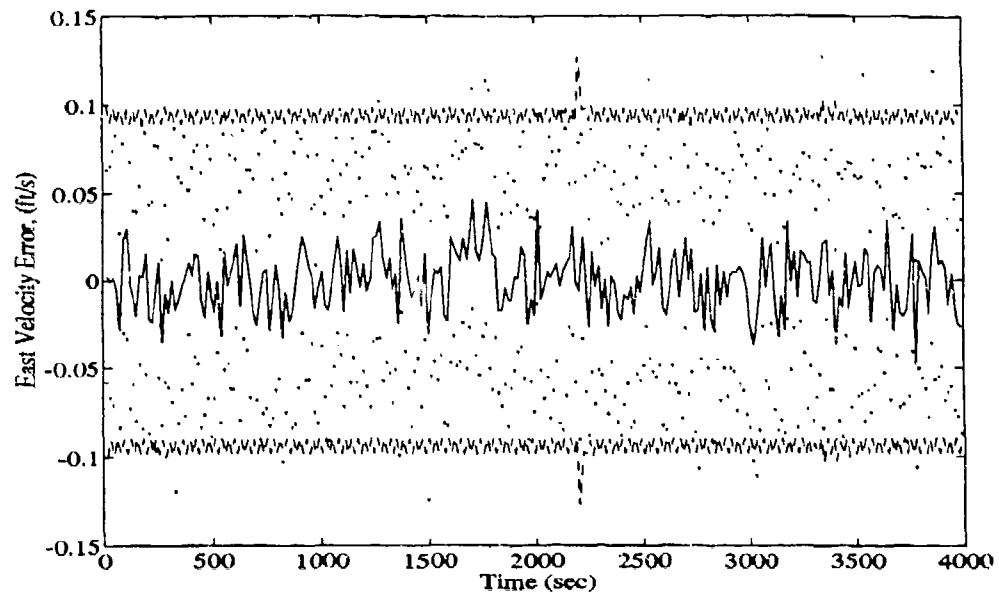


(a)

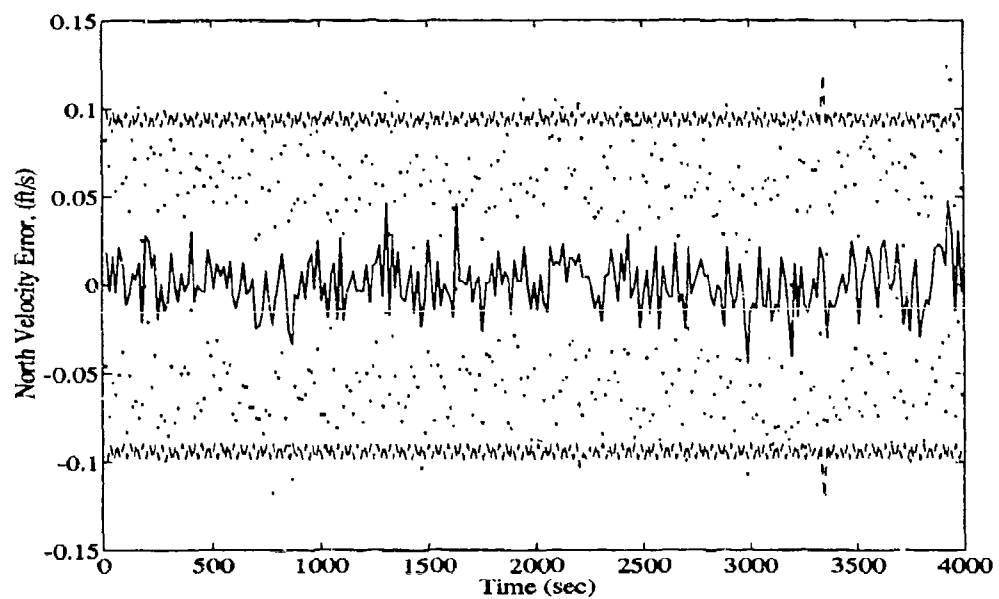


(b)

Figure C.3 West and North Tilt Error Filter Tuning Plots

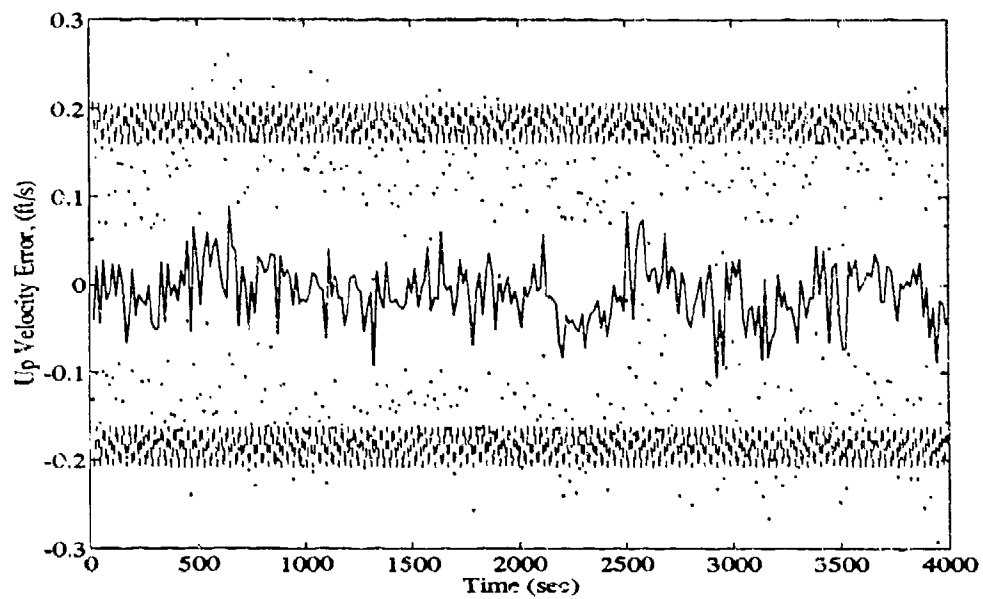


(a)

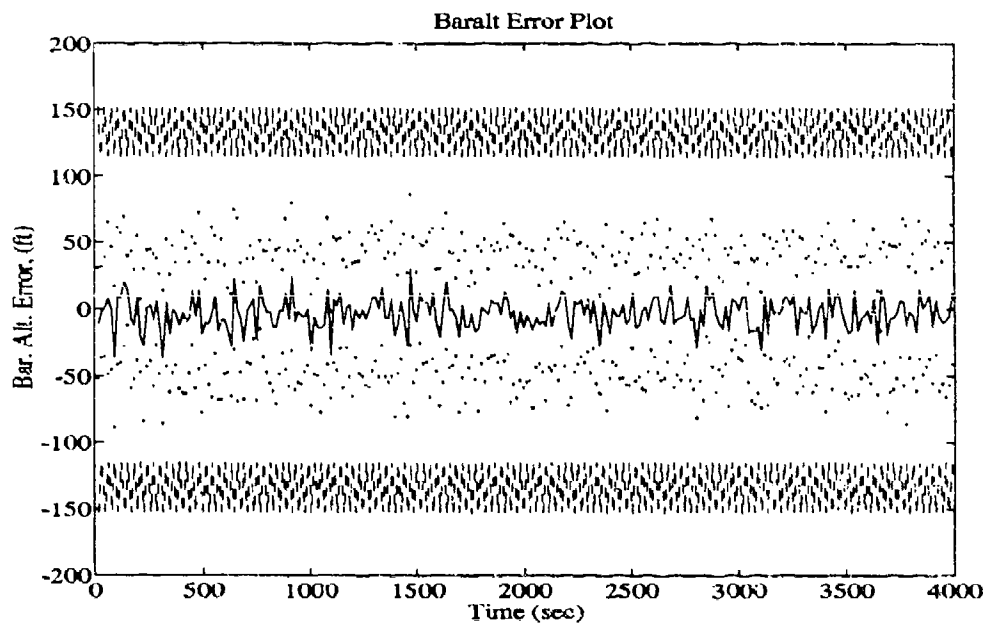


(b)

Figure C.4 North and West Velocity Error Filter Tuning Plots

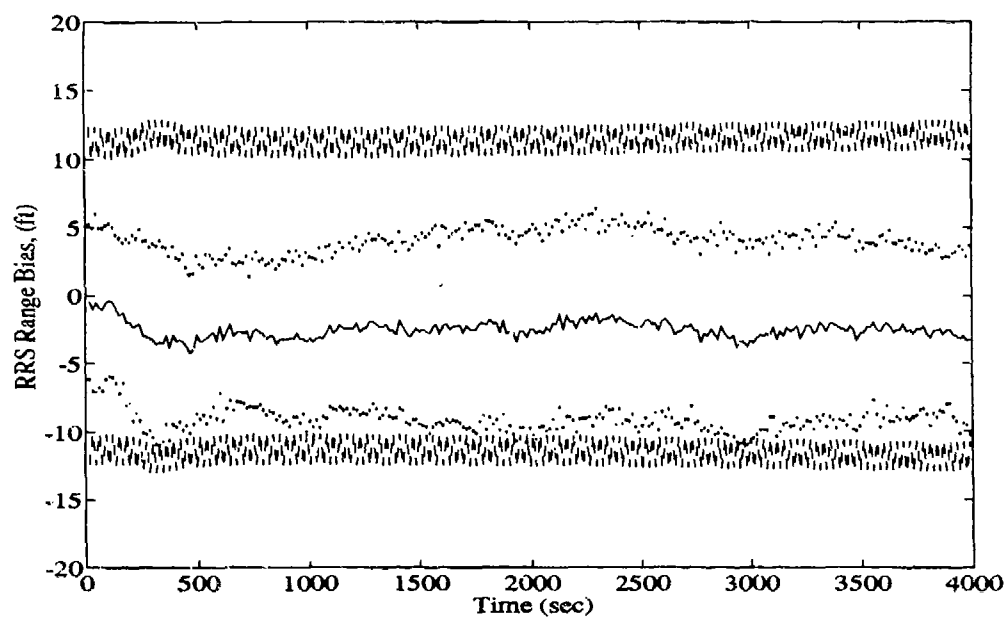


(a)

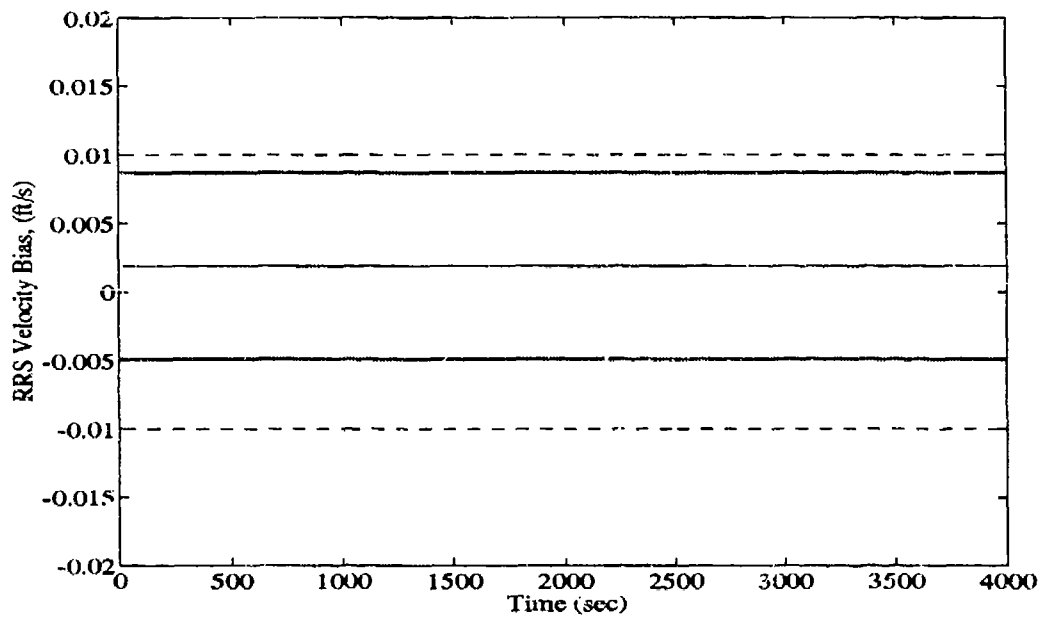


(b)

Figure C.5 Vertical Velocity and Barometric Altimeter Error Filter Tuning Plots

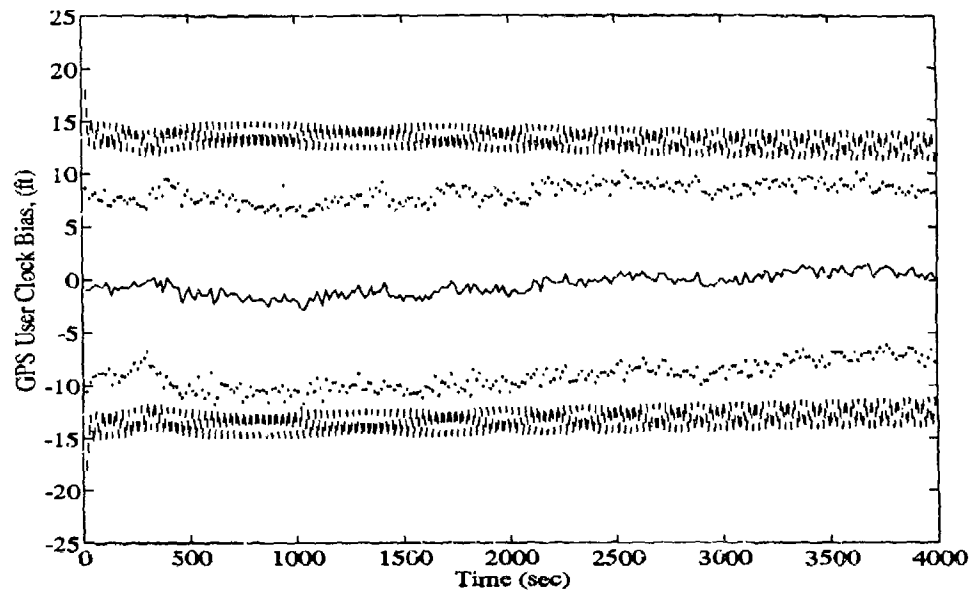


(a)

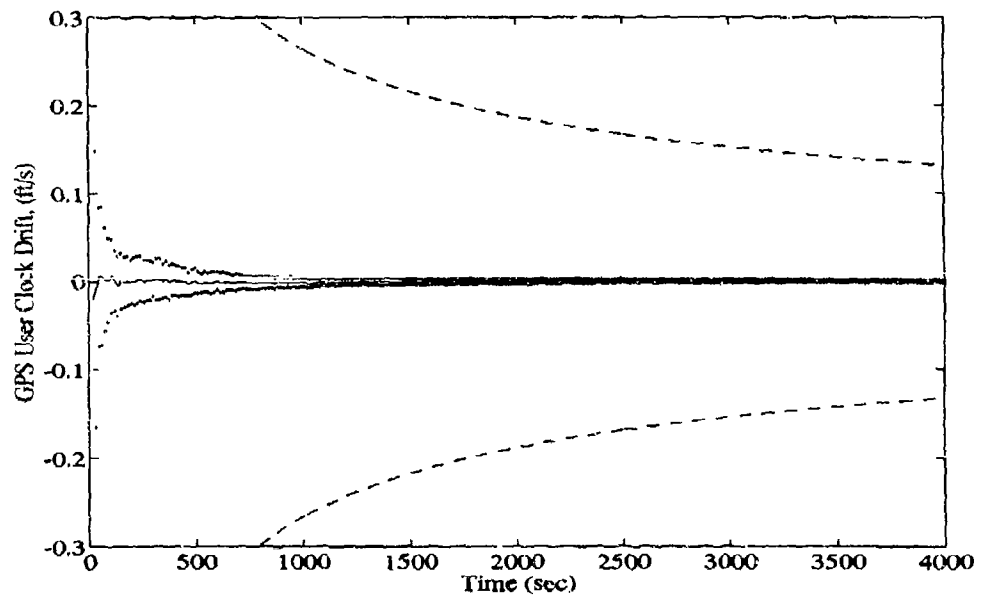


(b)

Figure C.6 RRS Range and Velocity Bias Filter Tuning Plots



(a)



(b)

Figure C.7 GPS User Clock Bias and Drift Filter Tuning Plots

Appendix D. Baseline Plots for Residual Monitoring

The GLR and Chi-Square algorithms implemented in this thesis rely heavily on the residuals produced by the individual NRS filters in the MNRS. The onset of failures in the system is reflected by the changes to the residuals of the extended Kalman filters. In this appendix, the residuals of the satellite and the transponder measurement updates are plotted for a failure-free run. These plots are used for comparison with the failure runs in the following appendices. These residual plots are similar in structure to the tuning plots of Appendix C. Each plot compares the Monte Carlo mean value of a residual, \pm the Monte Carlo calculated standard deviation for a scalar measurement residual, and zero \pm the filter-predicted standard deviation, . The derivation of the three plotted values is shown below.

The center solid line on the following plots is the mean value of a measurement residual. Similar to the plots in Appendix C, the mean value is defined as the Monte Carlo average of the residual over the time history of the filter run. The following equation is the mathematical definition of the mean value for a scalar residual [9, 13].

$$\hat{M}_\gamma(t_i) = \frac{1}{N} \sum_{j=1}^N \gamma_j(t_i) = \frac{1}{N} \sum_{j=1}^N \{z_j(t_i) - h_j[\hat{x}(t_i^-), t_i]\} \quad (D.1)$$

where $\gamma_j(t_i)$ is the scalar residual for the scalar measurement update $z_j(t_i)$. The subscript, j , refers to the j th Monte Carlo run. N is the total number of Monte Carlo runs in the simulation (15 in this thesis). $h_j[\hat{x}(t_i^-), t_i]$ represents the j th scalar vector of the $h[\hat{x}(t_i^-), t_i]$ vector that corresponds to the specific scalar measurement.

In addition to the center trace, two more *pairs* of lines are plotted. The first pair is the *Mean \pm Sigma* (represented by the \cdots lines in the plots). These lines are symmetrically displaced about the residual mean value, $\hat{M}_\gamma(t_i)$. The *Mean \pm Sigma* is the sum of the previously defined mean, $\hat{M}_\gamma(t_i)$, and the actual residual standard deviation $\pm\sqrt{P_\gamma(t_i)}$, where $P_\gamma(t_i)$ is the *true* error variance at time t_i . The true standard deviation is calculated

from the following equation [9, 19].

$$\sigma_{true}(t_i) = \sqrt{P_{\gamma}(t_i)} = \sqrt{\frac{1}{N-1} \sum_{j=1}^N \gamma_j^2(t_i) - \frac{N}{N-1} \hat{M}_{\gamma}^2(t_i)} \quad (D.2)$$

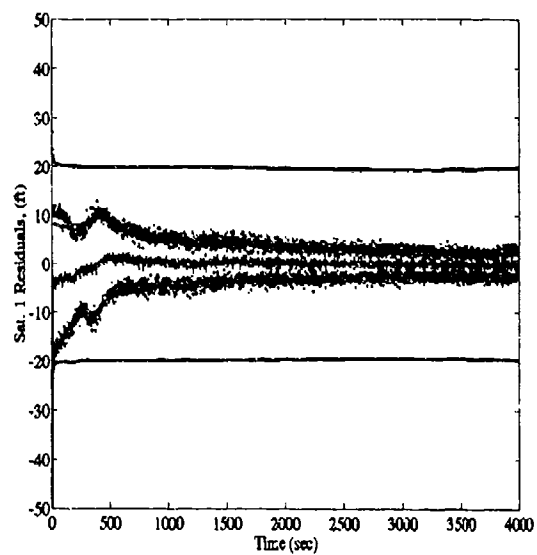
N is the number of runs in the Monte Carlo simulation, and $\hat{M}_{\gamma}^2(t_i)$ is the square of the calculated mean of a given residual at each time of interest.

The second pair of traces (represented by the - - lines) are the filter-computed $\pm \sigma_{filter}$ values for residual standard deviation [10, 19]. Equation (2.29) is used to calculate the full covariance matrix for the residuals. The scalar residual variances are the diagonal terms in the $[HP^{\top}H^T + R]$ residual covariance matrix. These plotted lines represent the filter's confidence in the scalar residual.

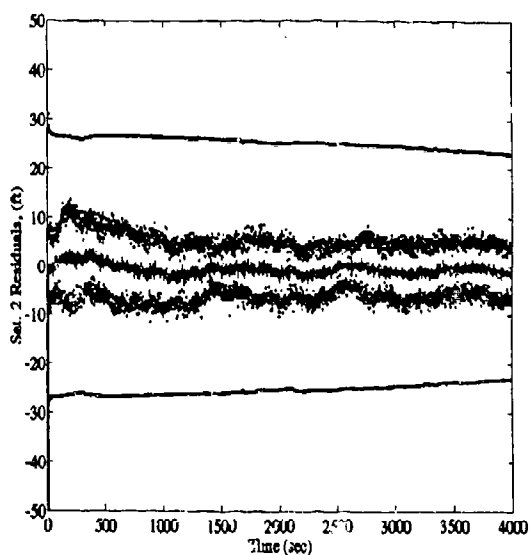
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. This section contains baseline, no-failure residual plots for the nine measurement updates (4 Satellites and 5 Transponders) in each of the *ten* NRS filters. The first plots presented are for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS10. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table D.1 Legend for Filter Tuning Plots

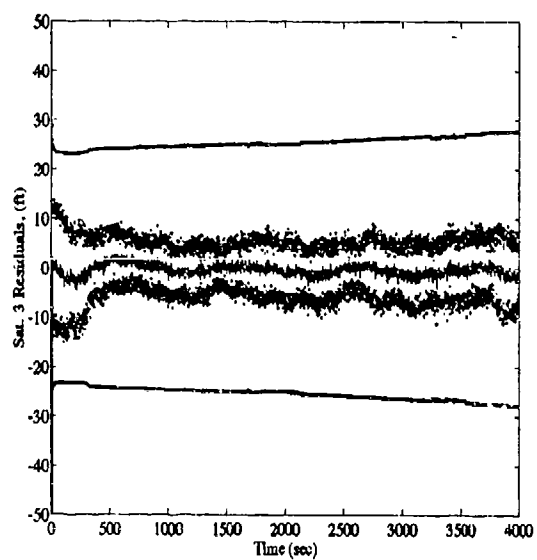
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
- - Dashed Line	\pm Filter-Predicted Sigma



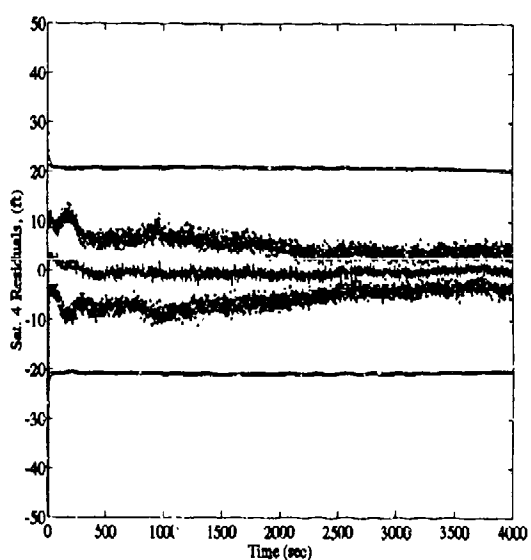
(Sat 1)



(Sat 2)

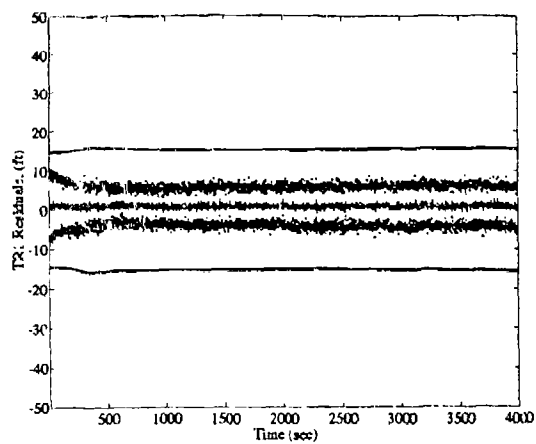


(Sat 3)

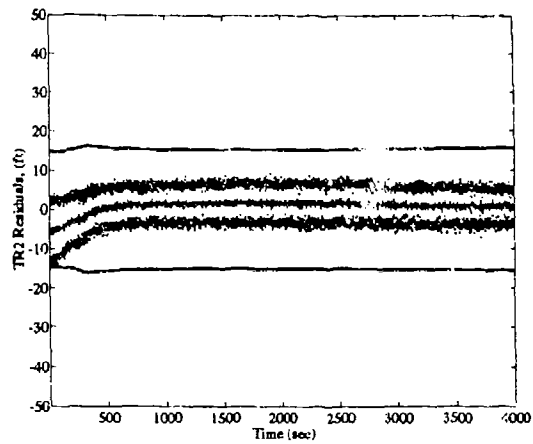


(Sat 4)

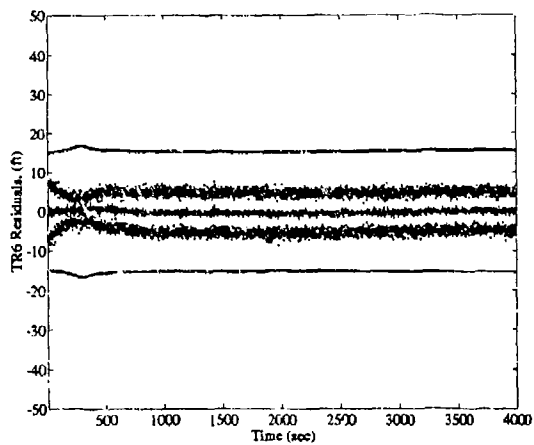
Figure D.1 NRS1 Satellite Scalar Residual Plots, Baseline



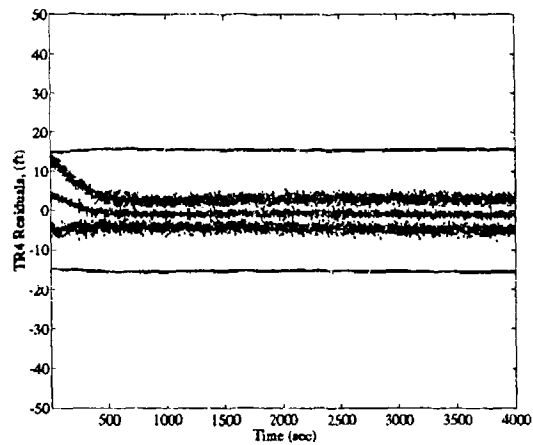
(Trans 1)



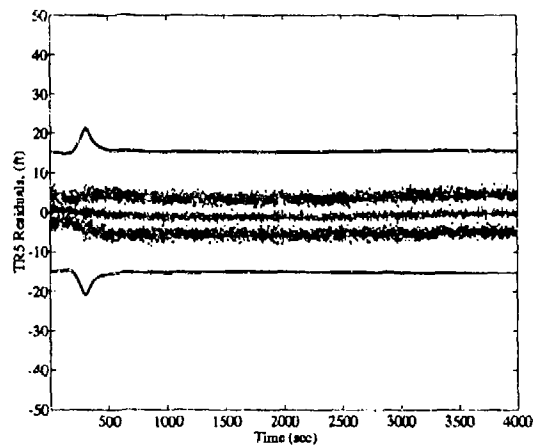
(Trans 2)



(Trans 3)

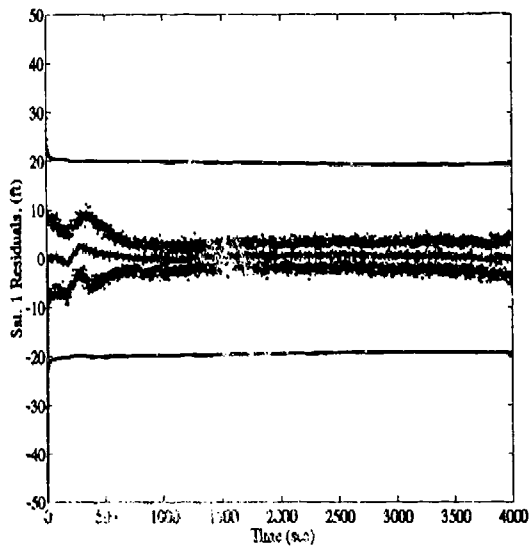


(Trans 4)

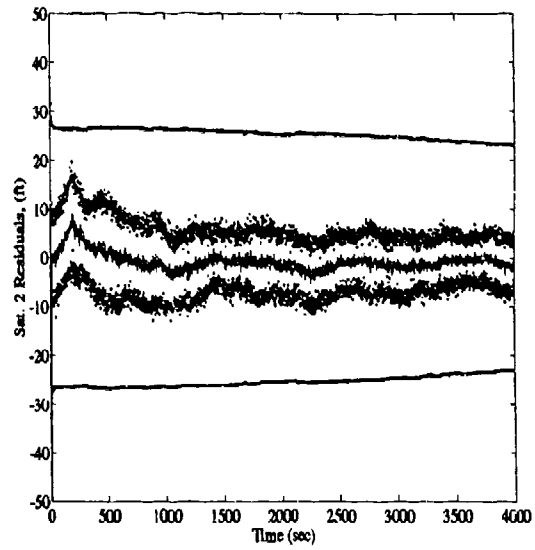


(Trans 5)

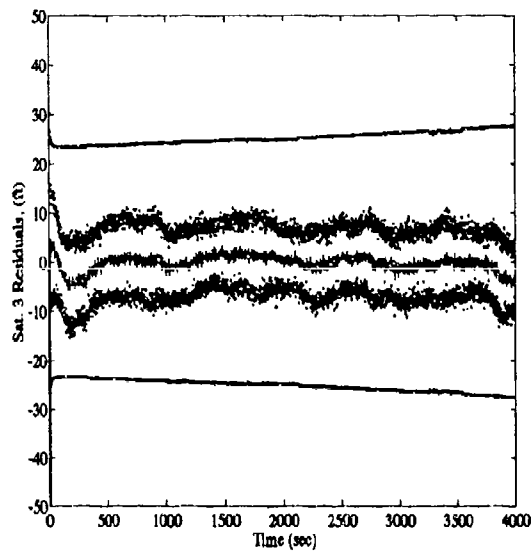
Figure D.2 NRS1 Transponder Scalar Residual Plots, Baseline



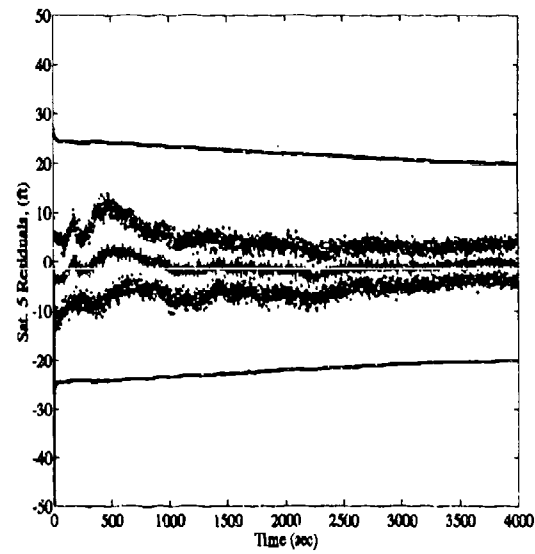
(Sat 1)



(Sat 2)

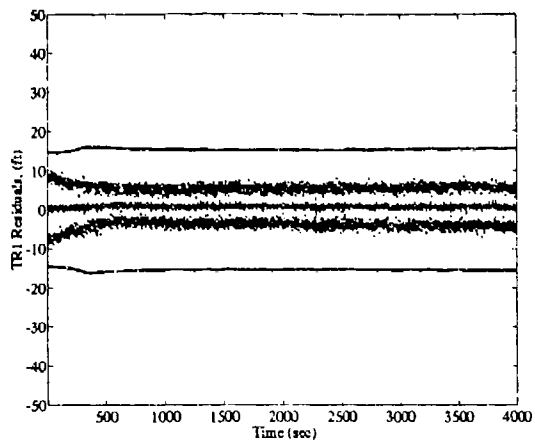


(Sat 3)

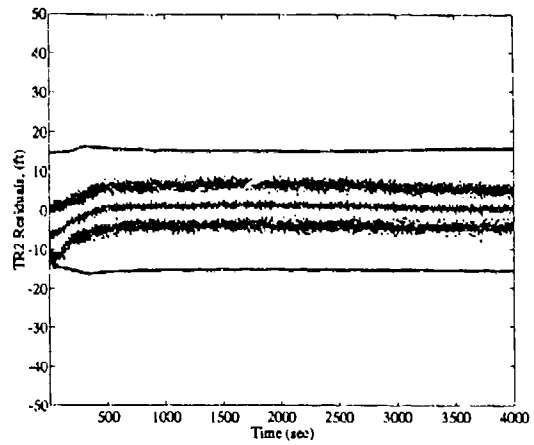


(Sat 5)

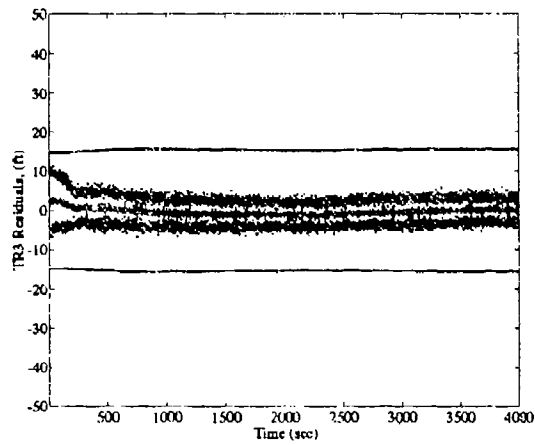
Figure D.3 NRS2 Satellite Scalar Residual Plots, Baseline



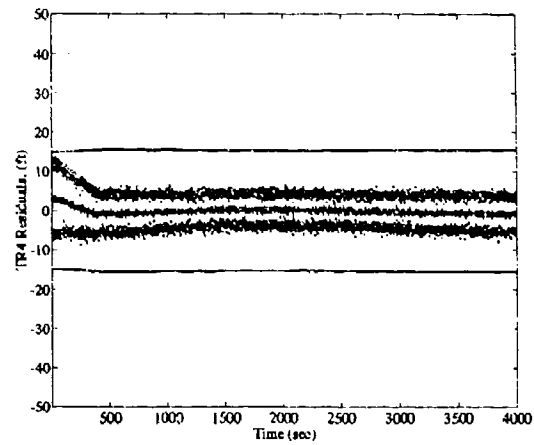
(Trans 1)



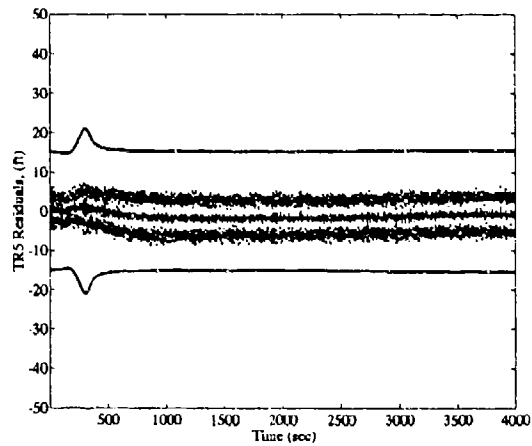
(Trans 2)



(Trans 3)

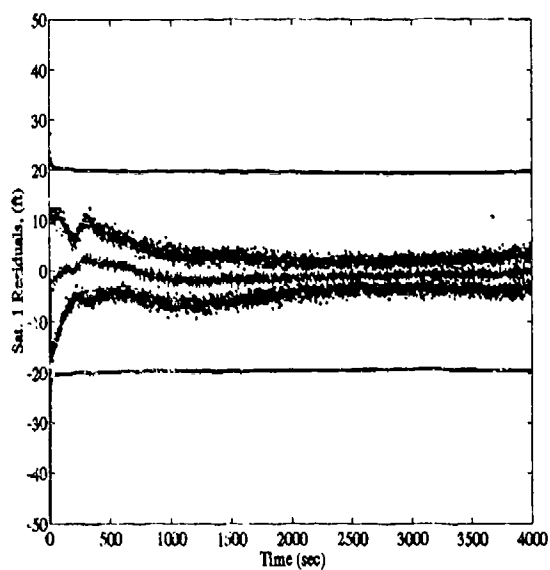


(Trans 4)

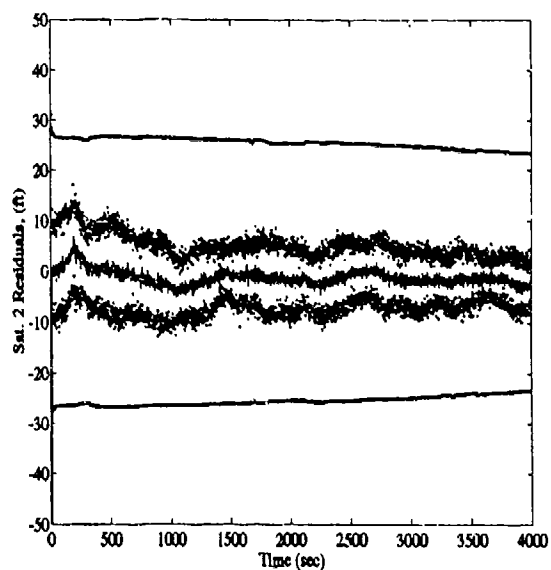


(Trans 5)

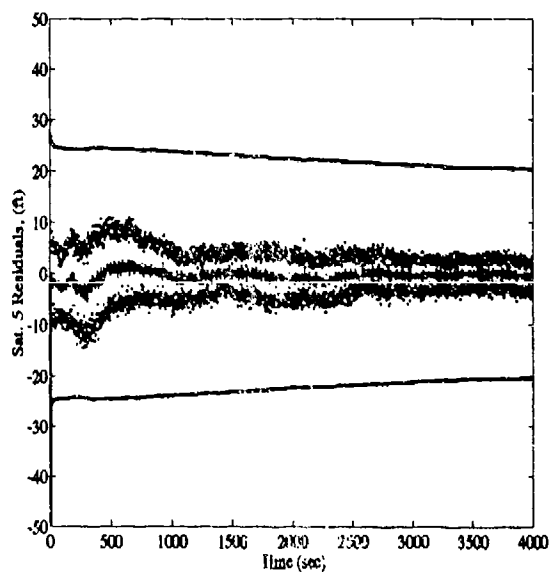
Figure D.4 NRS2 Transponder Scalar Residual Plots, Baseline



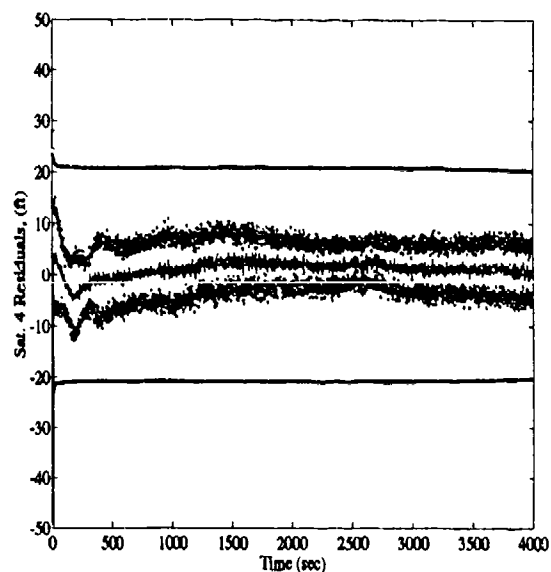
(Sat 1)



(Sat 2)

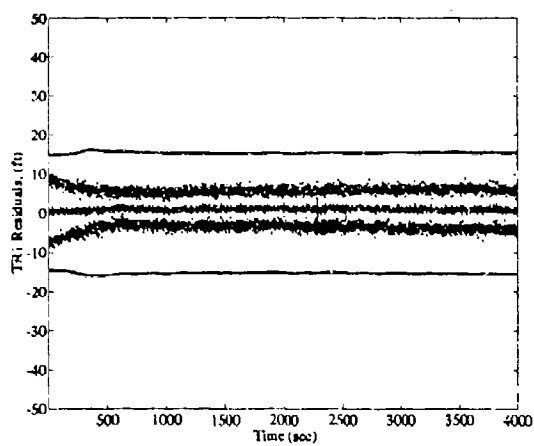


(Sat 5)

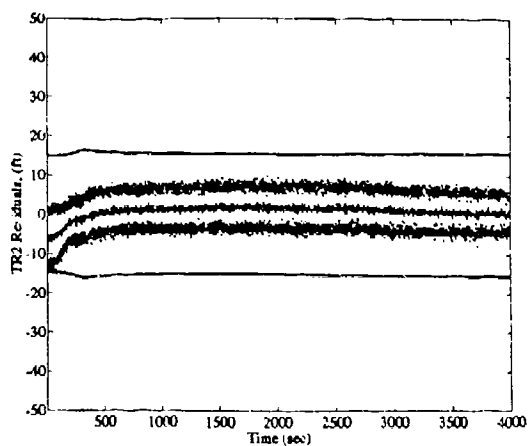


(Sat 4)

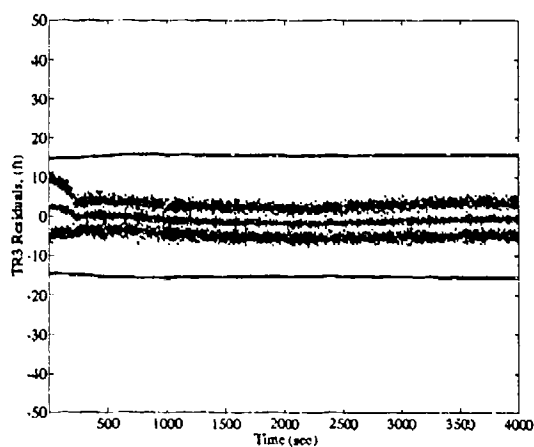
Figure D.5 NRS3 Satellite Scalar Residual Plots, Baseline



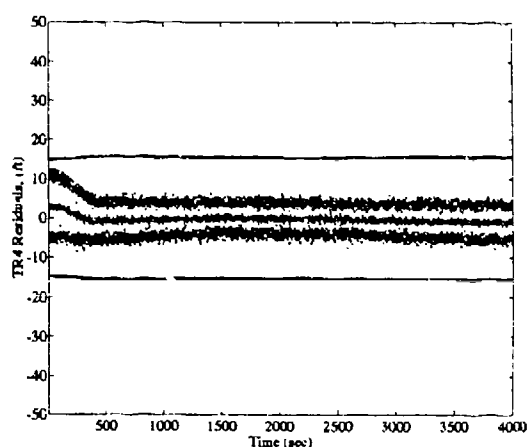
(Trans 1)



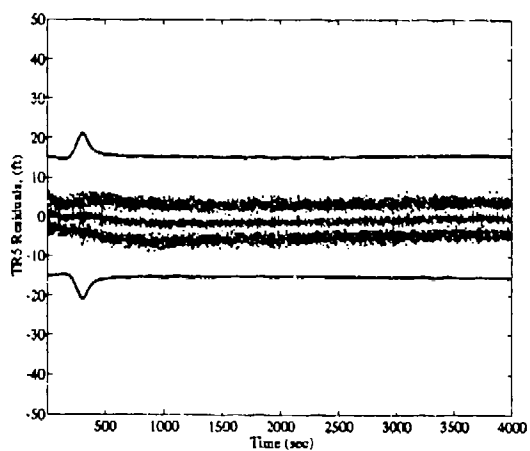
(Trans 2)



(Trans 3)

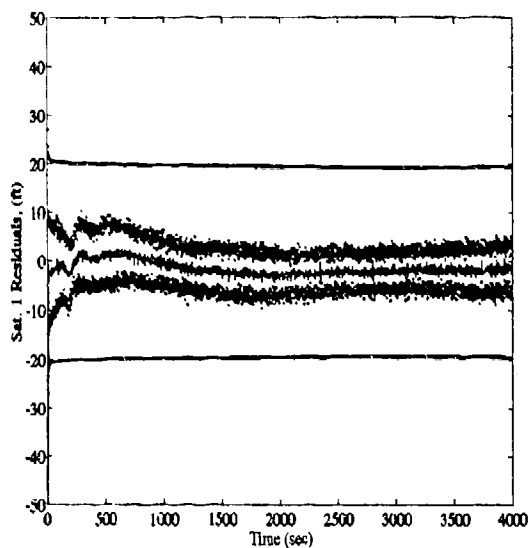


(Trans 4)

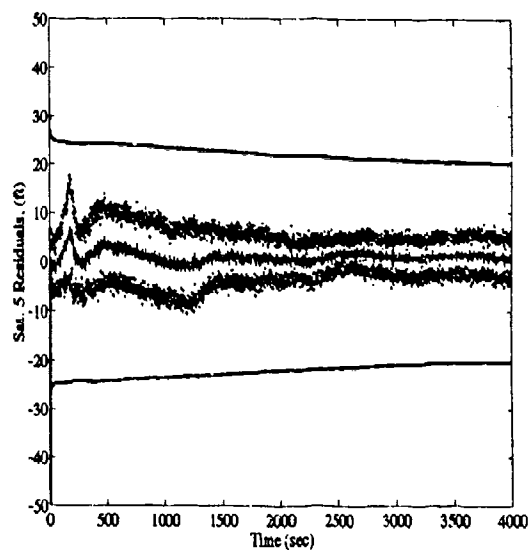


(Trans 5)

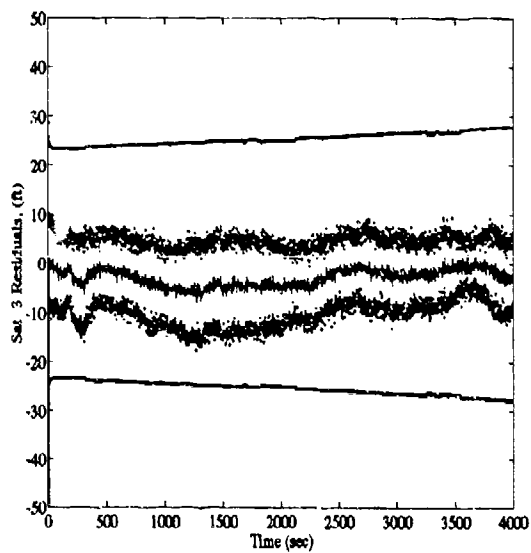
Figure D.6 NRS3 Transponder Scalar Residual Plots, Baseline



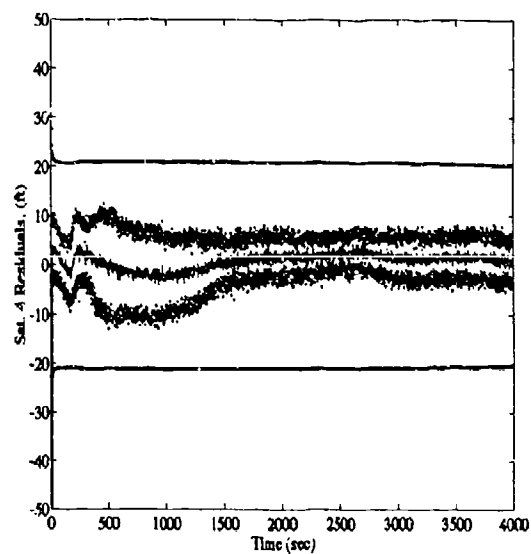
(Sat 1)



(Sat 5)

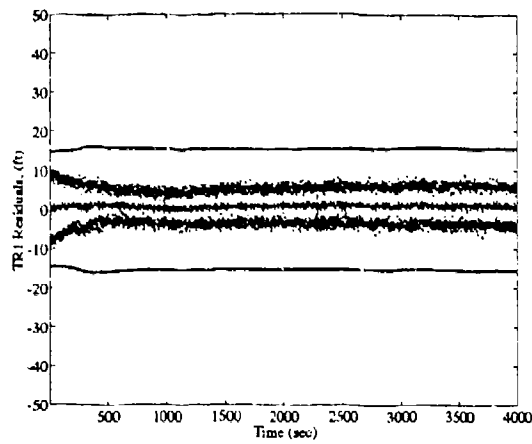


(Sat 3)

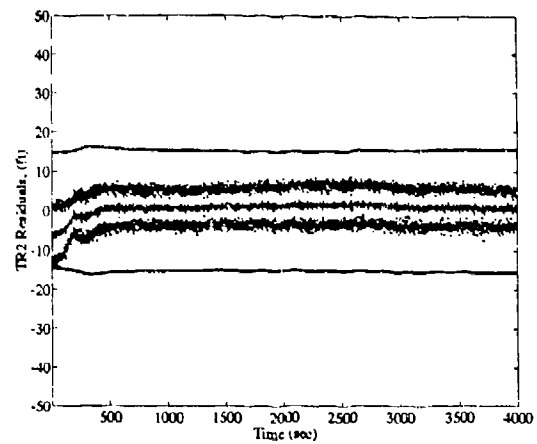


(Sat 4)

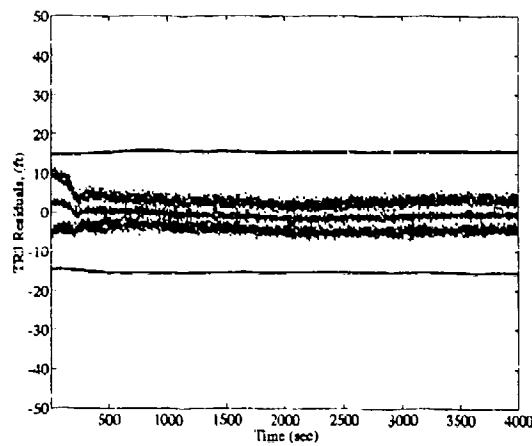
Figure D.7 NRS4 Satellite Scalar Residual Plots, Baseline



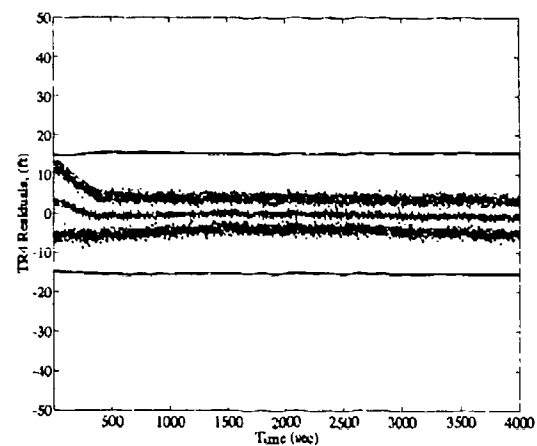
(Trans 1)



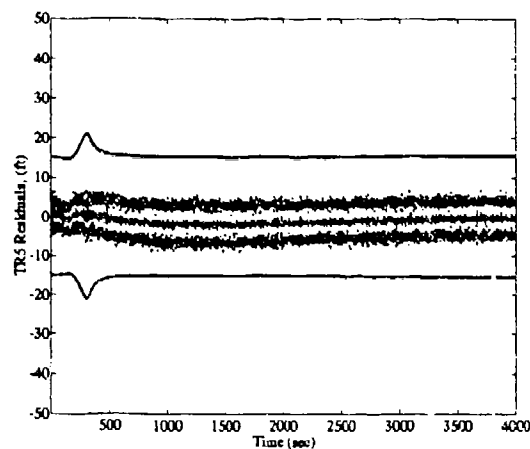
(Trans 2)



(Trans 3)

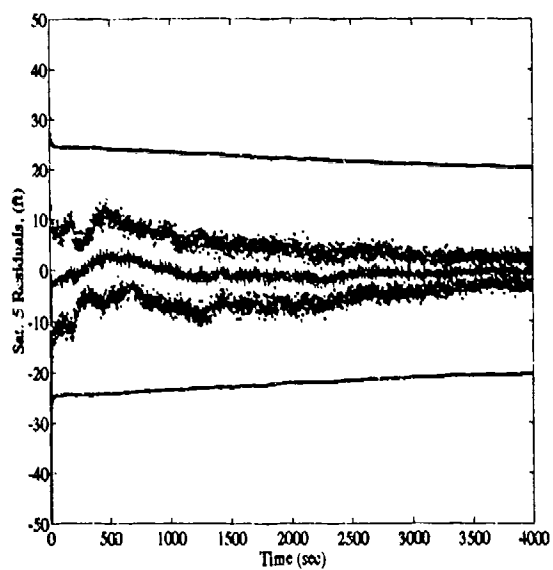


(Trans 4)

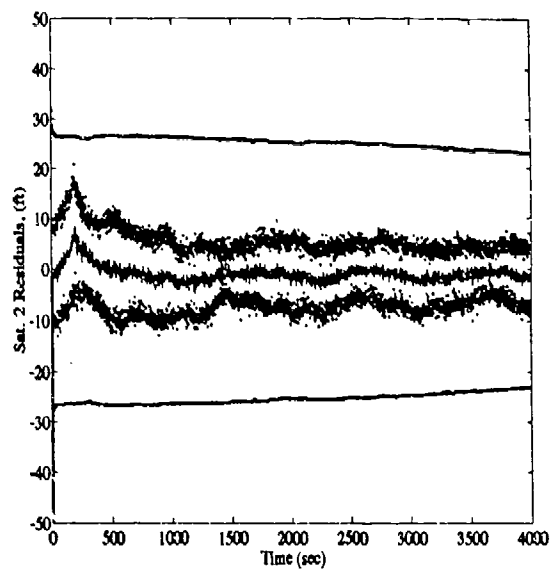


(Trans 5)

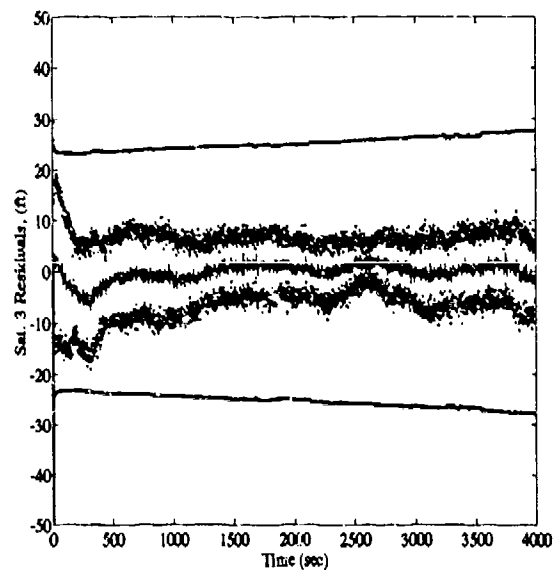
Figure D.8 NRS4 Transponder Scalar Residual Plots, Baseline



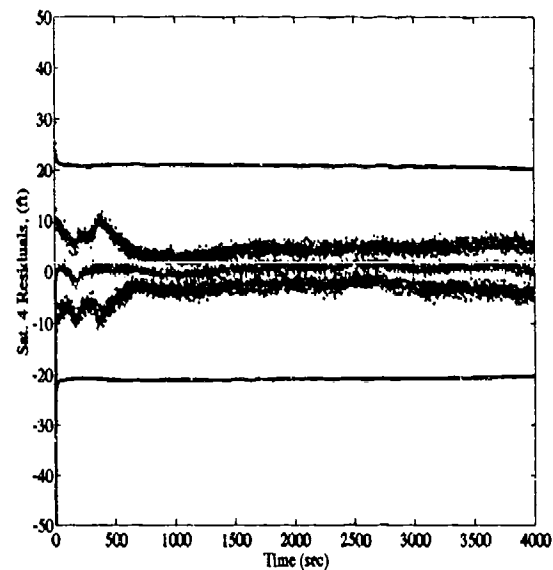
(Sat 5)



(Sat 2)

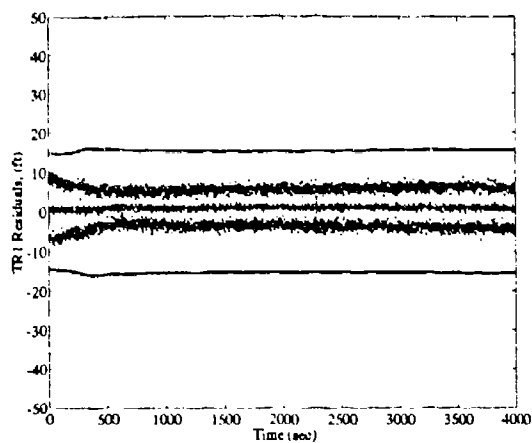


(Sat 3)

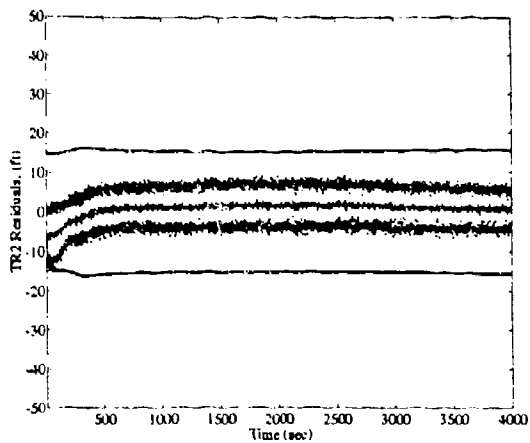


(Sat 4)

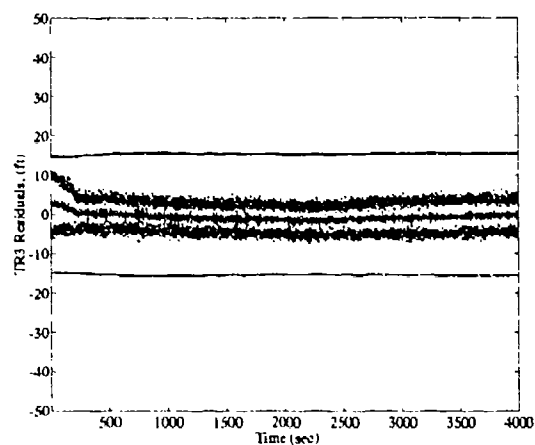
Figure D.9 NRS5 Satellite Scalar Residual Plots, Baseline



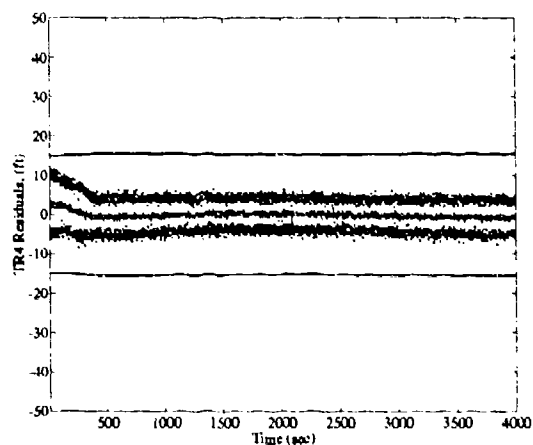
(Trans 1)



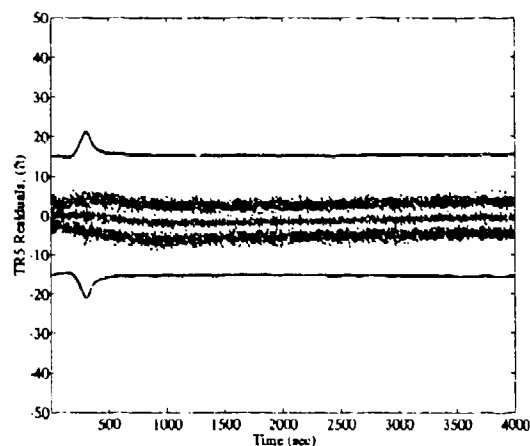
(Trans 2)



(Trans 3)

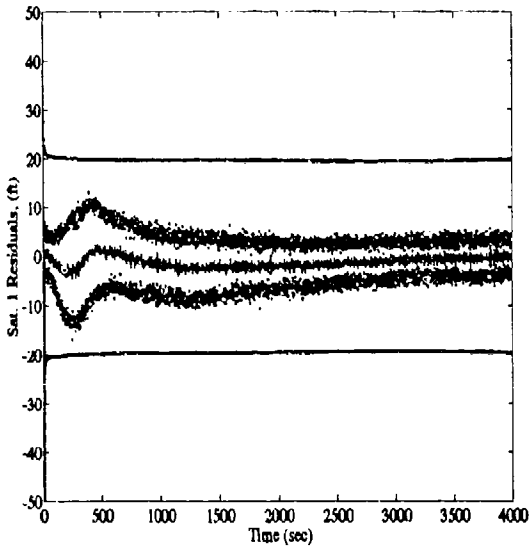


(Trans 4)

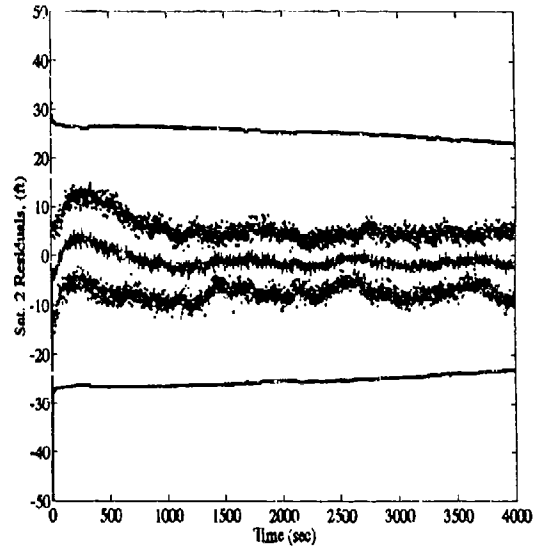


(Trans 5)

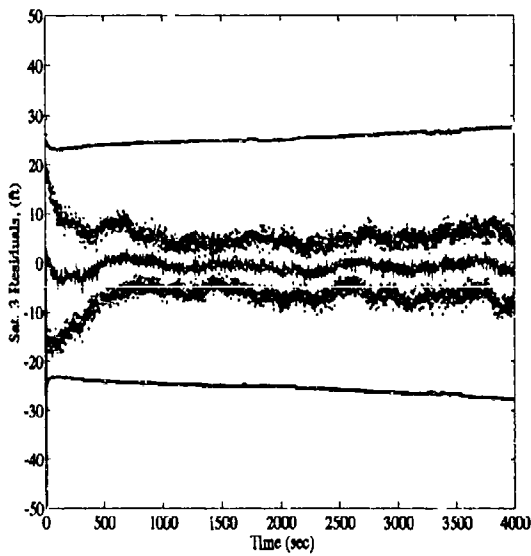
Figure D.10 NRS5 Transponder Scalar Residual Plots, Baseline



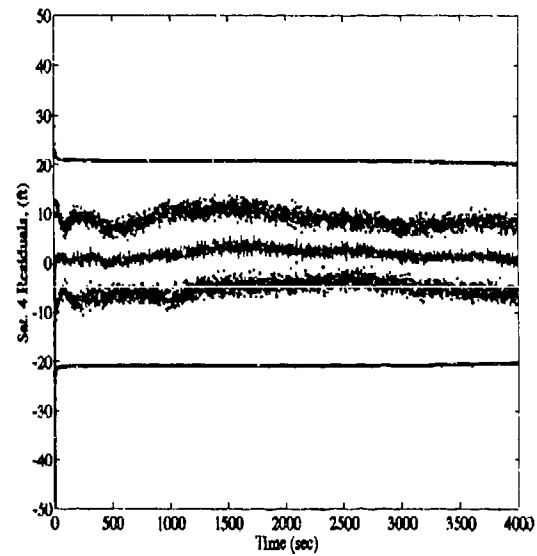
(Sat 1)



(Sat 2)

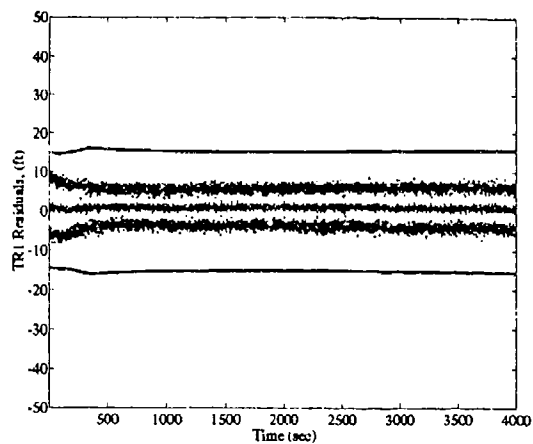


(Sat 3)

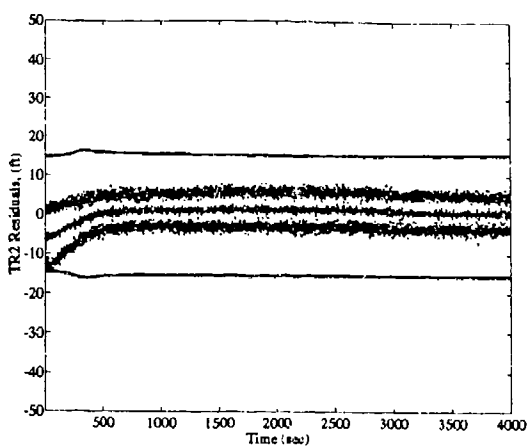


(Sat 4)

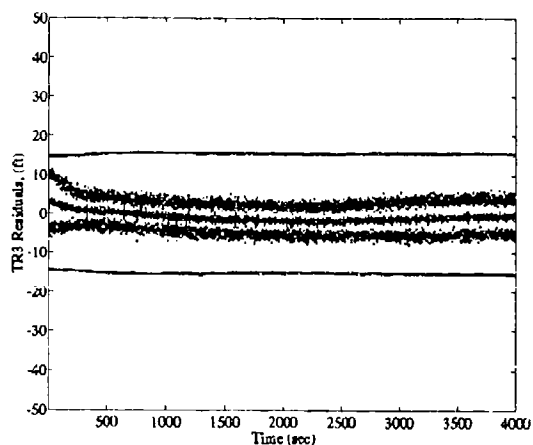
Figure D.11 NRS6 Satellite Scalar Residual Plots, Baseline



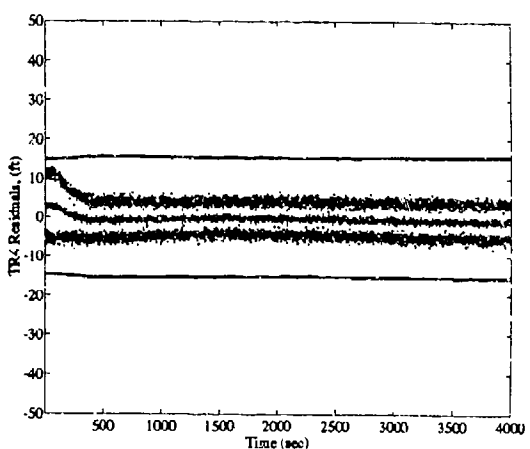
(Trans 1)



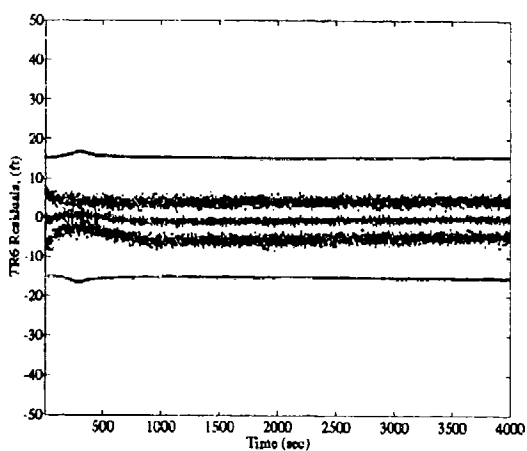
(Trans 2)



(Trans 3)

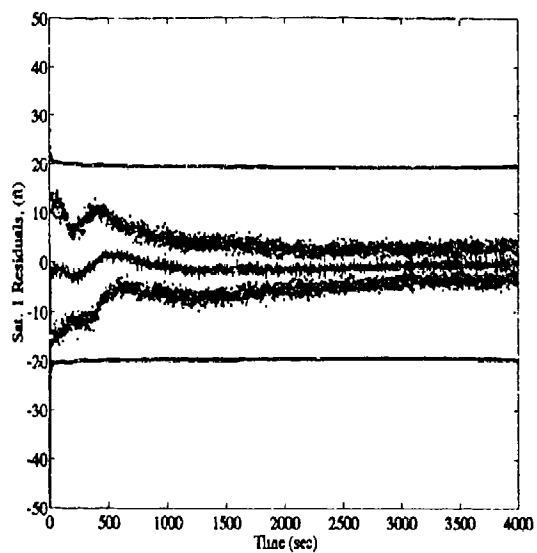


(Trans 4)

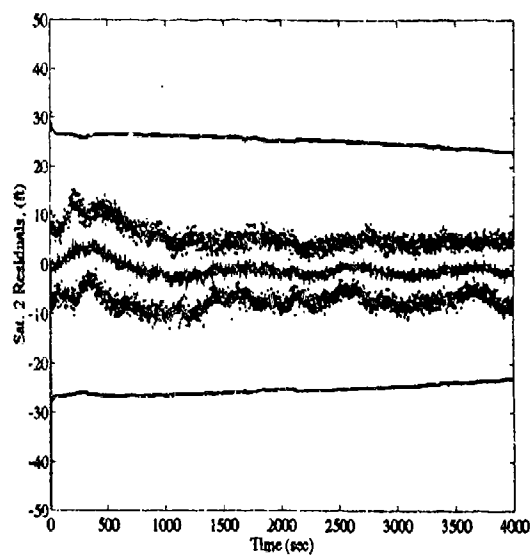


(Trans 6)

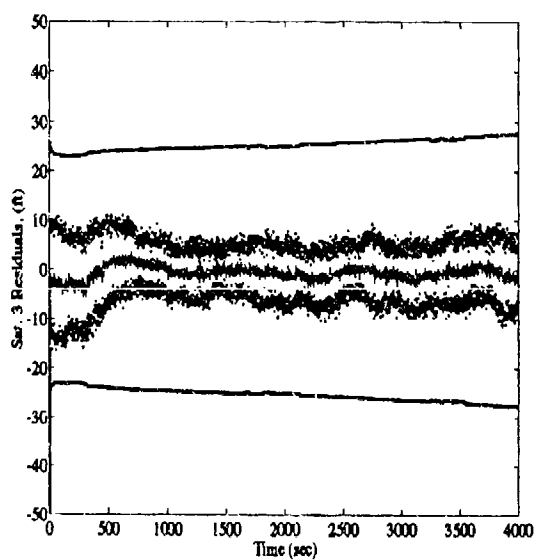
Figure D.12 NRS6 Transponder Scalar Residual Plots, Baseline



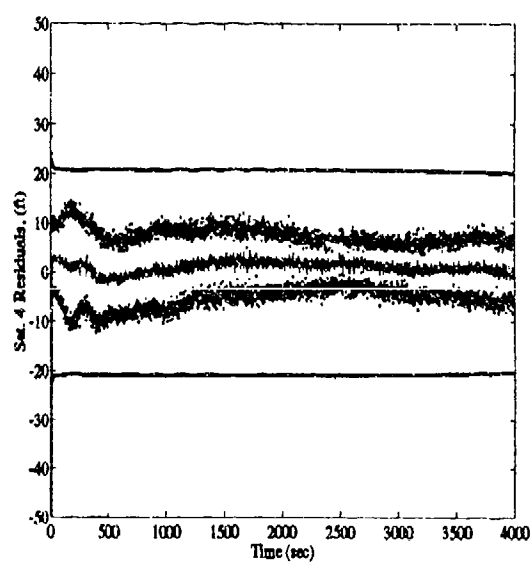
(Sat 1)



(Sat 2)

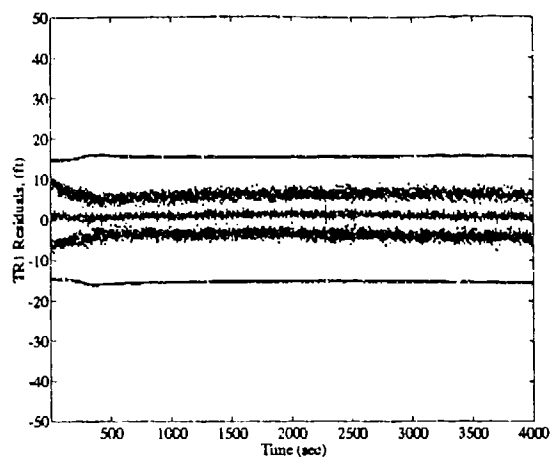


(Sat 3)

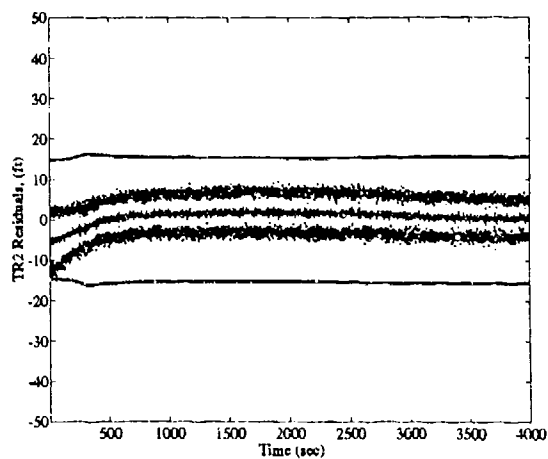


(Sat 4)

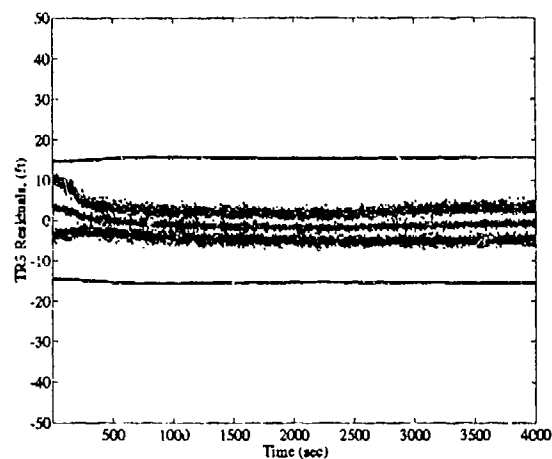
Figure D.13 NRS7 Satellite Scalar Residual Plots, Baseline



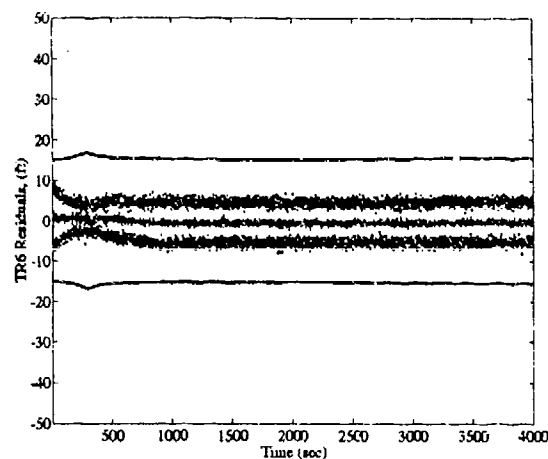
(Trans 1)



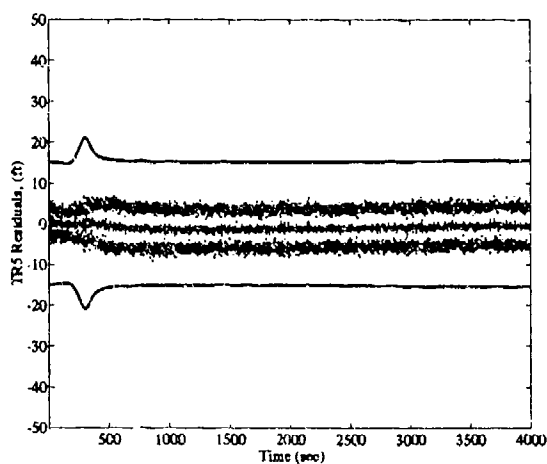
(Trans 2)



(Trans 4)

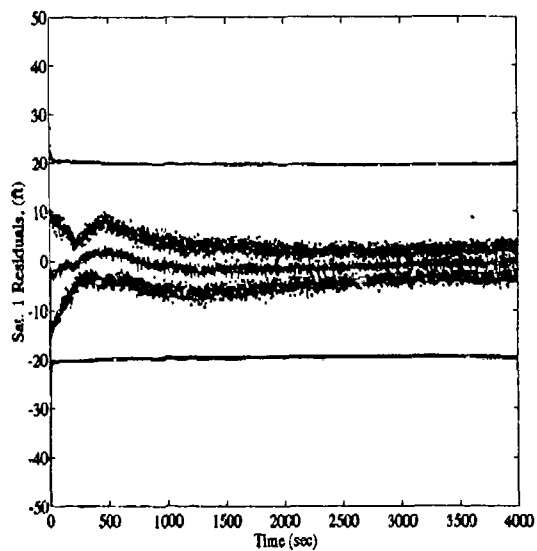


(Trans 6)

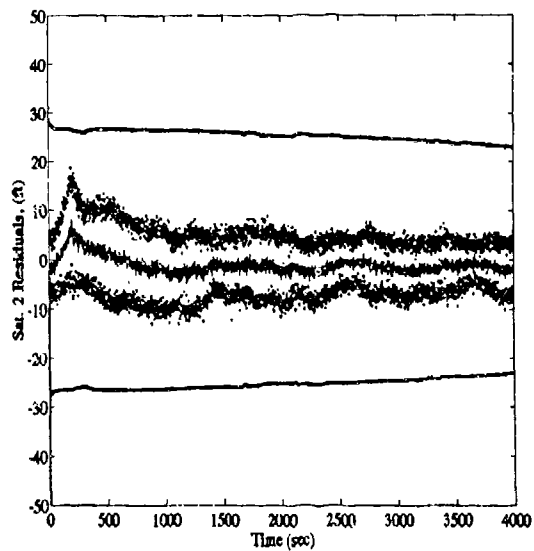


(Trans 5)

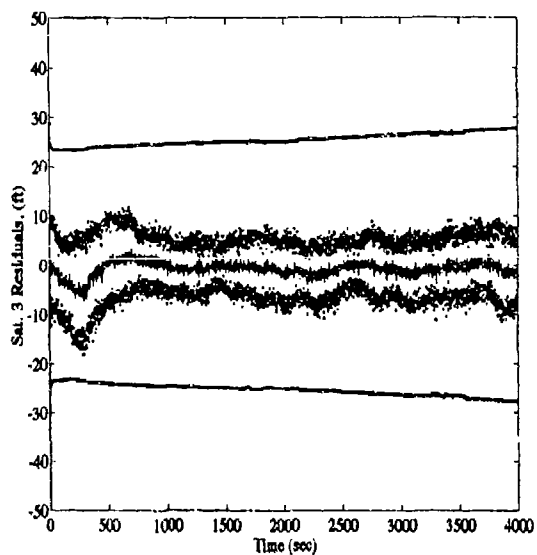
Figure D.14 NRS7 Transponder Scalar Residual Plots, Baseline



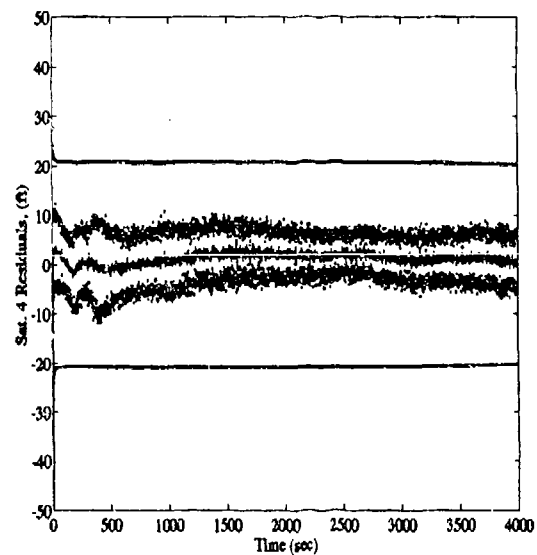
(Sat 1)



(Sat 2)

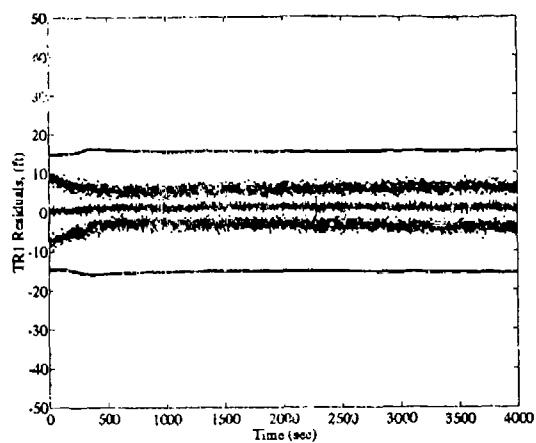


(Sat 3)

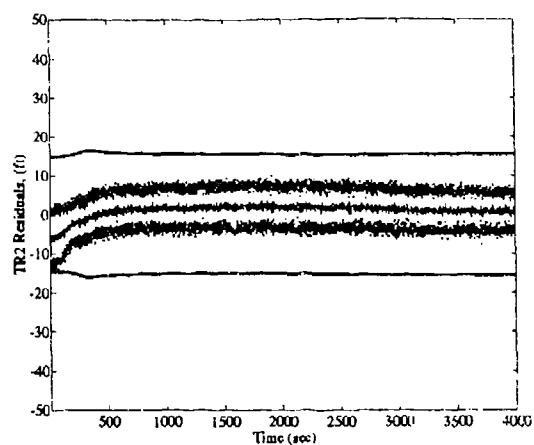


(Sat 4)

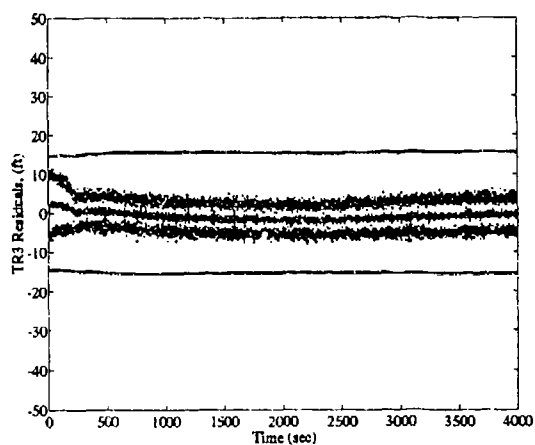
Figure D.15 NRS8 Satellite Scalar Residual Plots, Baseline



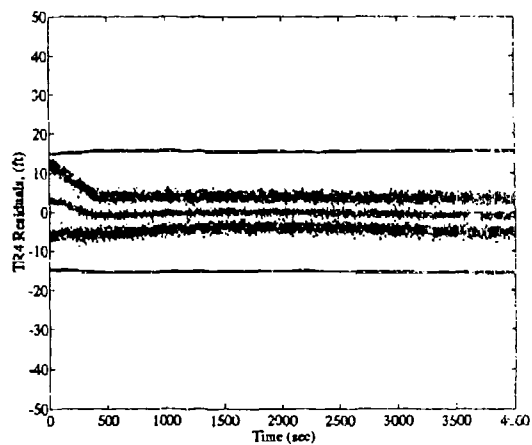
(Trans 1)



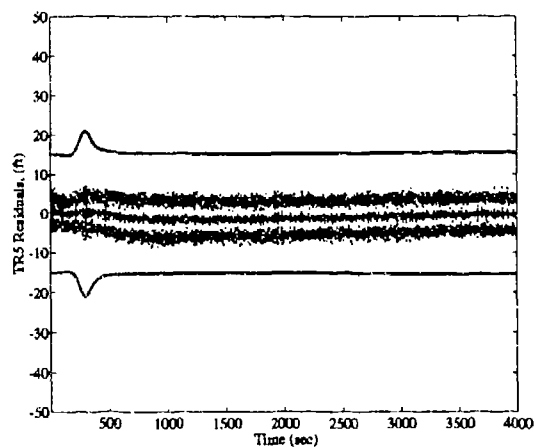
(Trans 2)



(Trans 6)

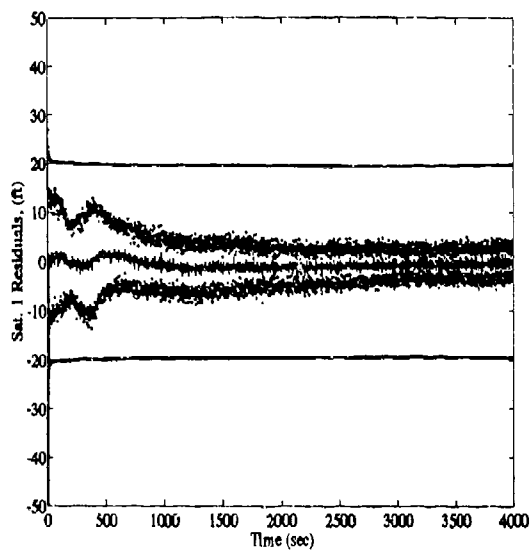


(Trans 4)

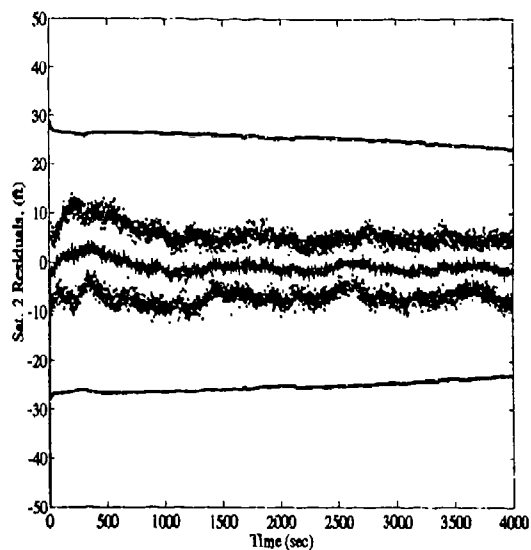


(Trans 5)

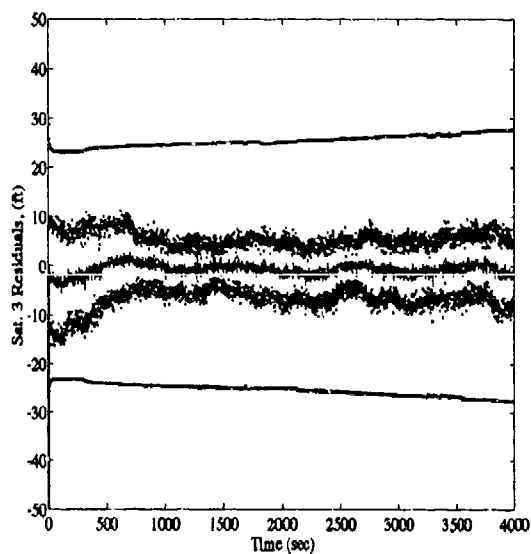
Figure D.16 NRS8 Transponder Scalar Residual Plots, Baseline



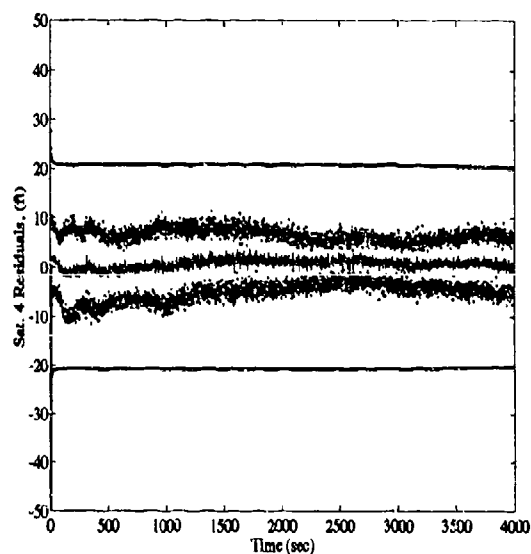
(Sat 1)



(Sat 2)

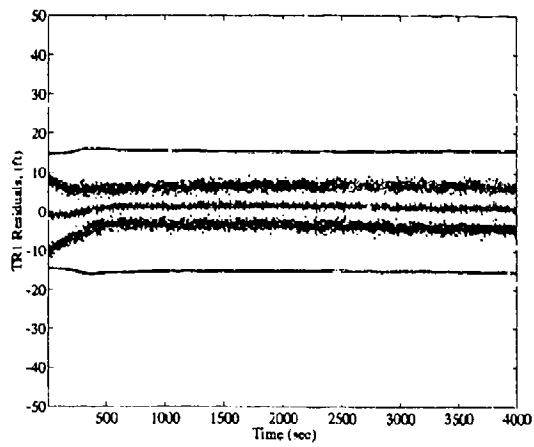


(Sat 3)

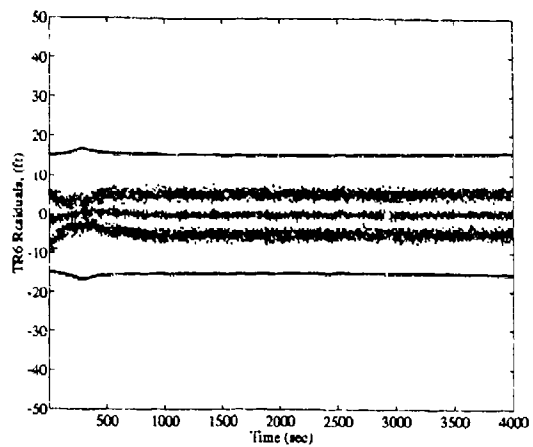


(Sat 4)

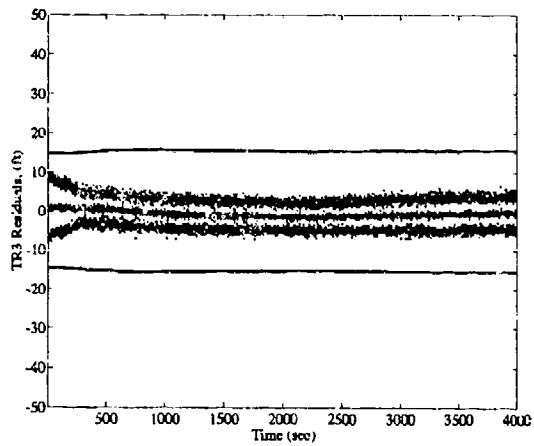
Figure D.17 NRS9 Satellite Scalar Residual Plots, Baseline



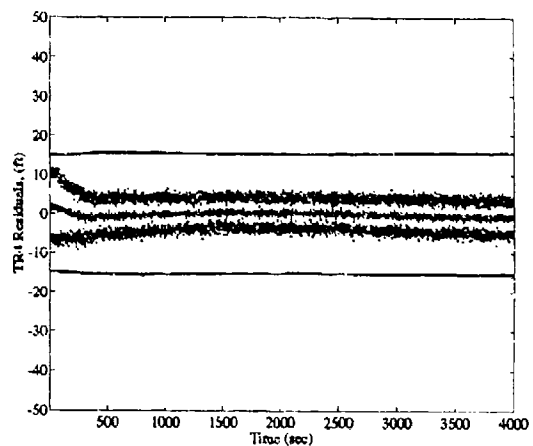
(Trans 1)



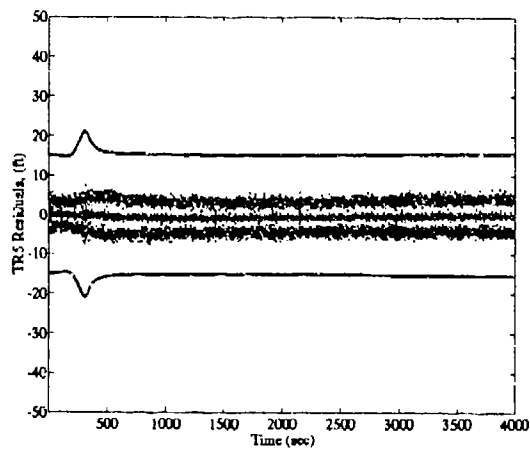
(Trans 6)



(Trans 3)

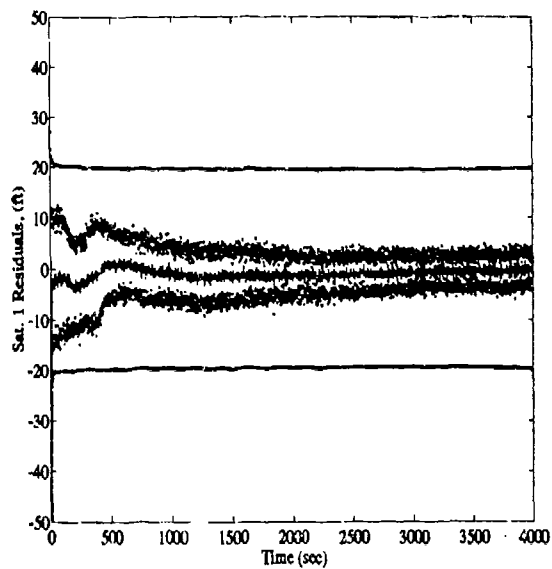


(Trans 4)

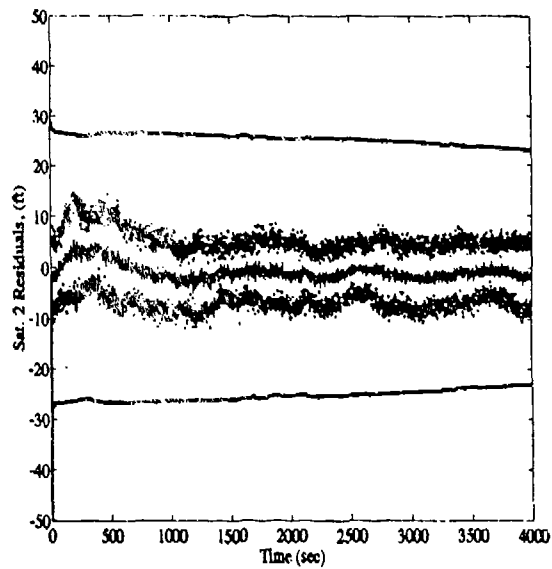


(Trans 5)

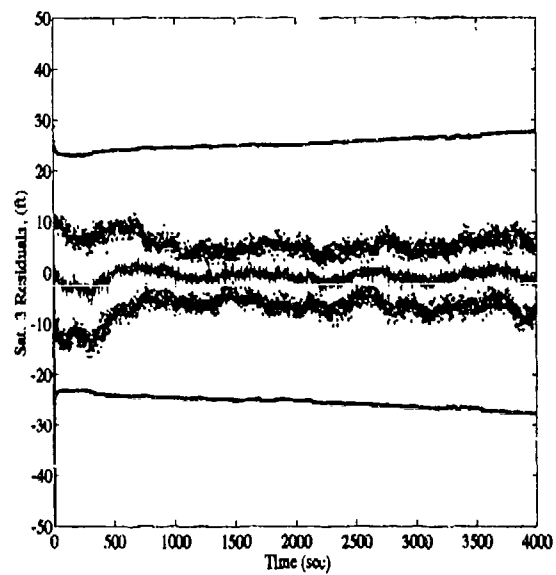
Figure D.18 NRS9 Transponder Scalar Residual Plots, Baseline



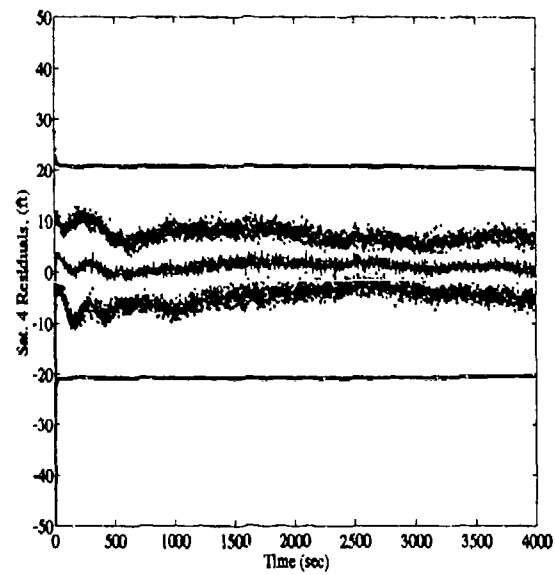
(Sat 1)



(Sat 2)

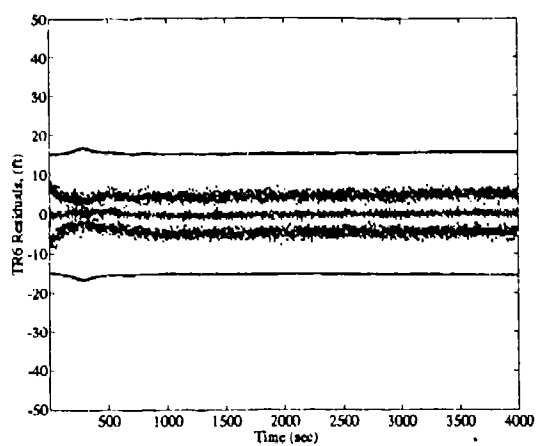


(Sat 3)

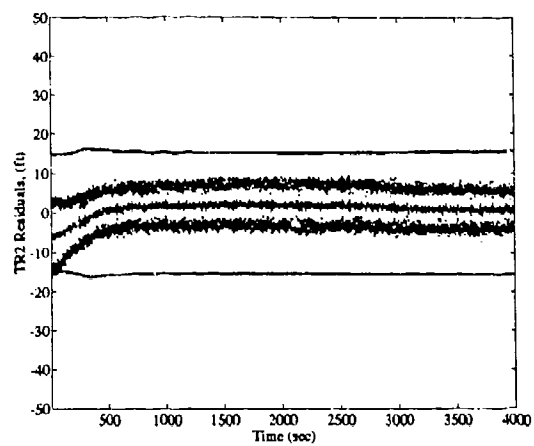


(Sat 4)

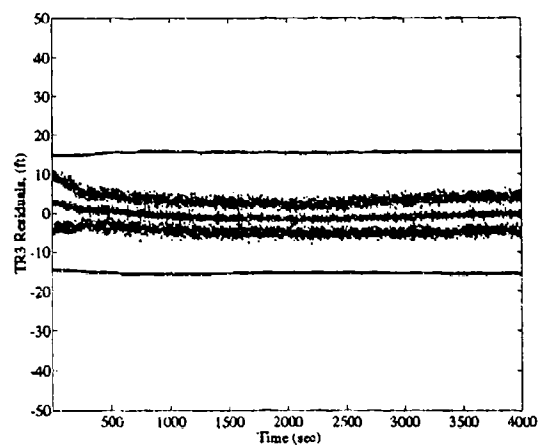
Figure D.19 NRS10 Satellite Scalar Residual Plots, Baseline



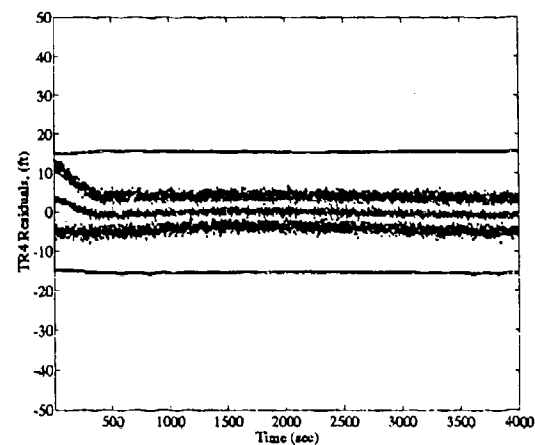
(Trans 6)



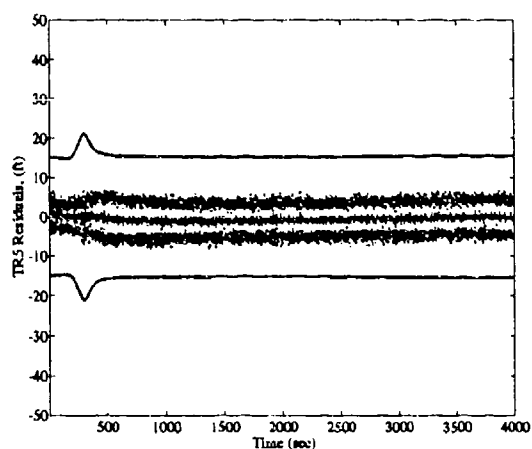
(Trans 2)



(Trans 3)



(Trans 4)



(Trans 5)

Figure D.20 NRS10 Transponder Scalar Residual Plots, Baseline

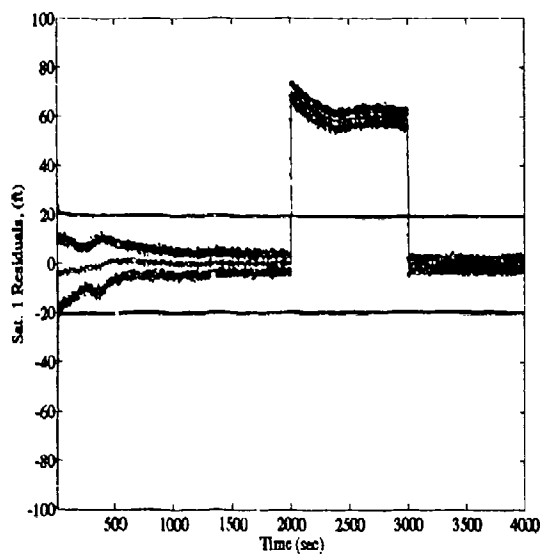
Appendix E. Residual Plots for a Transponder Signal Step Failure

This appendix contains all the scalar residual data for the ten NRS filters during a simulated step bias of 800ft on the Transponder 1 range signal from time $t = 2000sec$ through $t = 3000sec$. The plots contained in this section are identical in format as those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

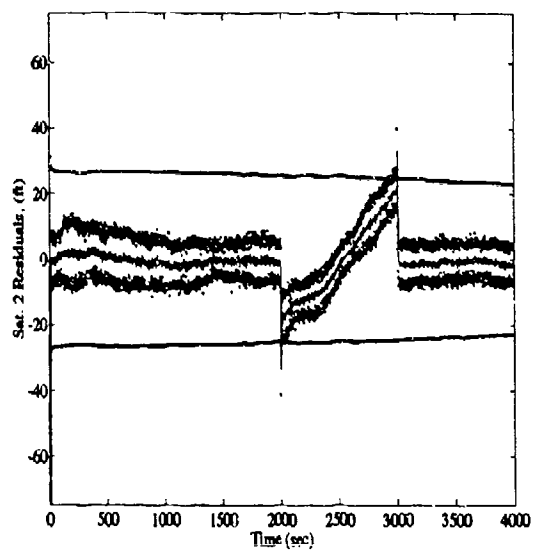
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. This section contains the plots for the nine measurement updates (4 Satellites and 5 Transponders) in each of the *ten* NRS filters. The first plots presented are for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS10. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table E.1 Legend for Filter Tuning Plots

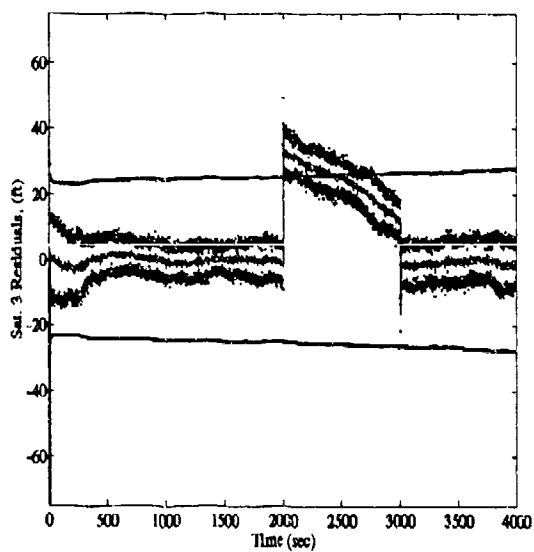
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
-- Dashed Line	\pm FilterPredicted Sigma



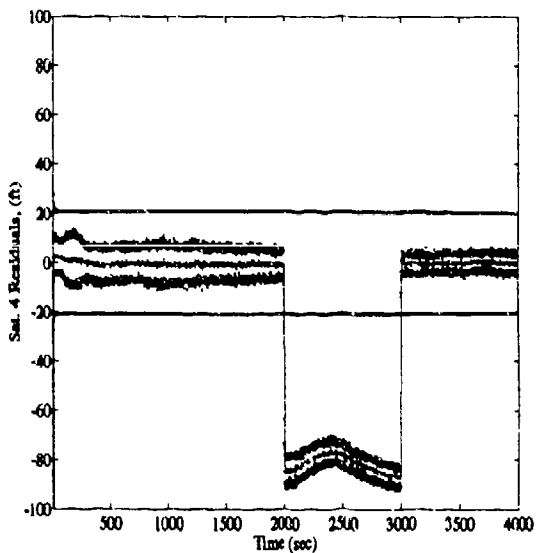
(Sat 1)



(Sat 2)

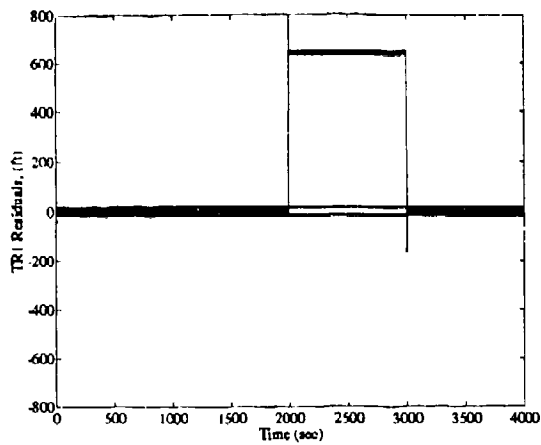


(Sat 3)

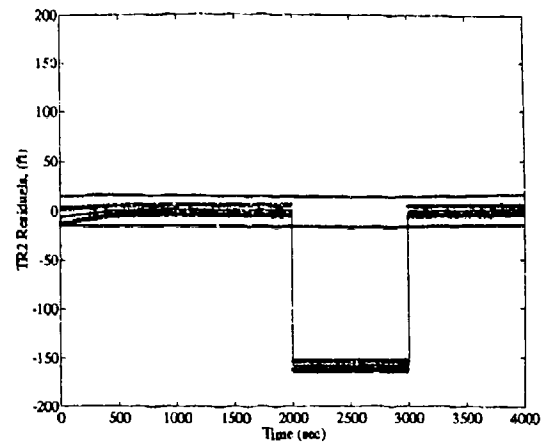


(Sat 4)

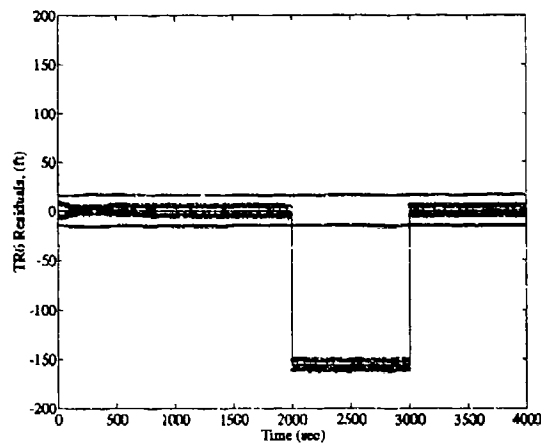
Figure E.1 NRS1 Satellite Scalar Residual Plots, Transponder Step Bias



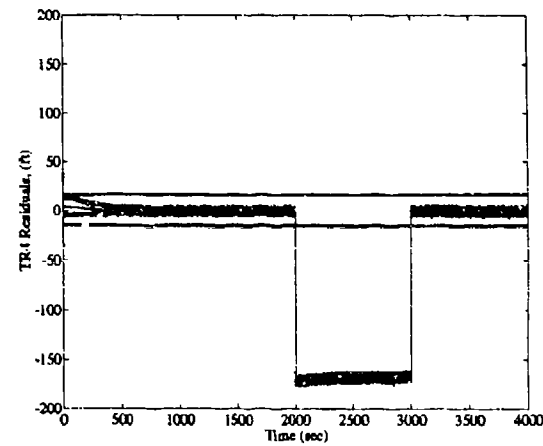
(Trans 1)



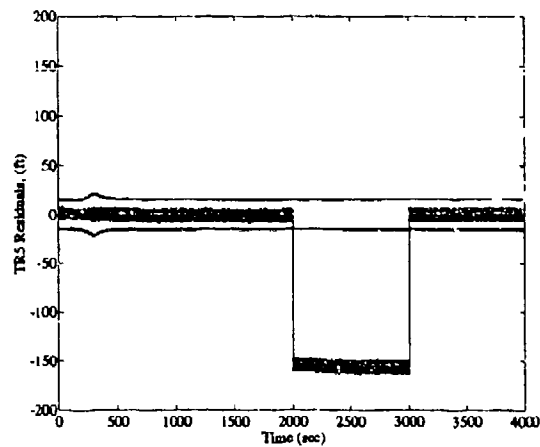
(Trans 2)



(Trans 3)

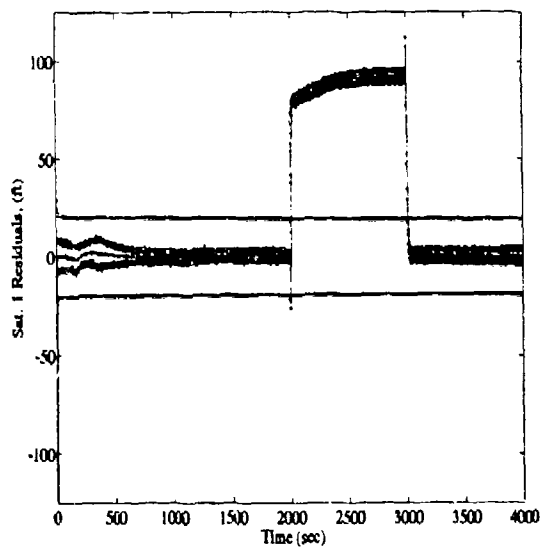


(Trans 4)

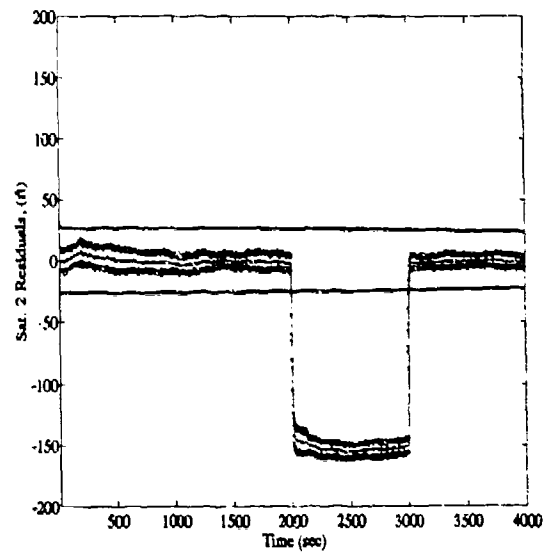


(Trans 5)

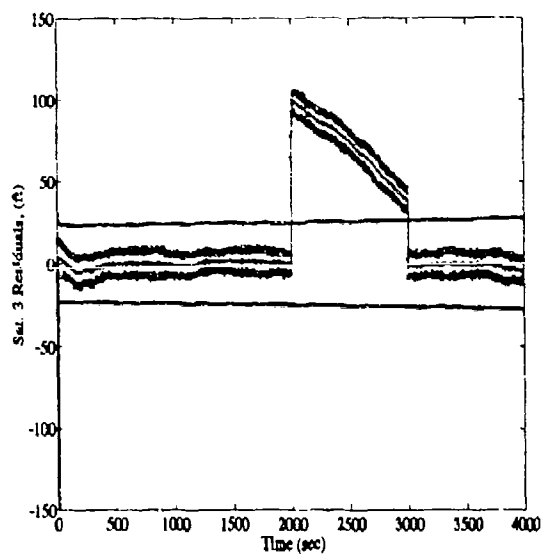
Figure E.2 NRS1 Transponder Scalar Residual Plots, Transponder Step Bias



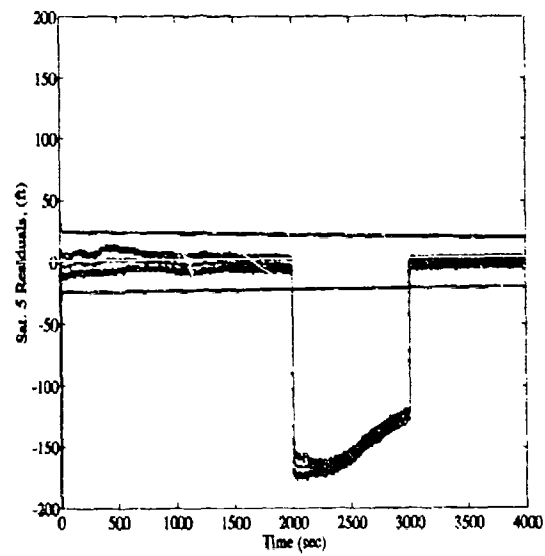
(Sat 1)



(Sat 2)

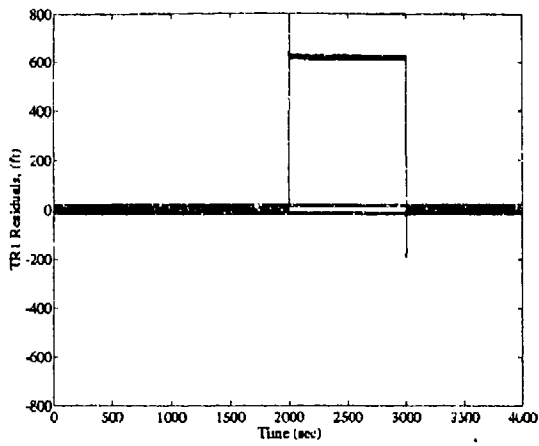


(Sat 3)

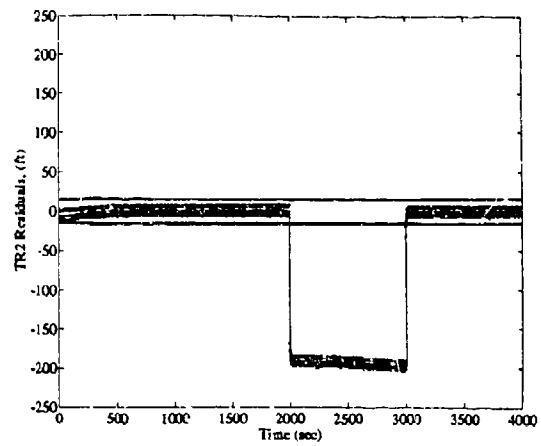


(Sat 5)

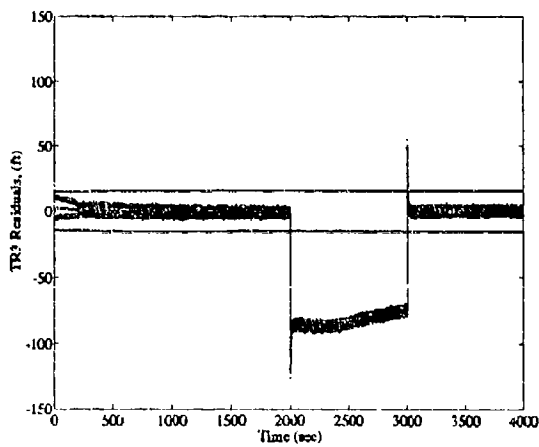
Figure E.3 NRS2 Satellite Scalar Residual Plots, Transponder Ramp Bias



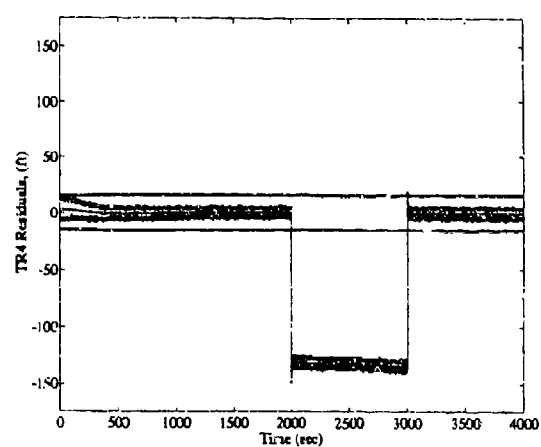
(Trans 1)



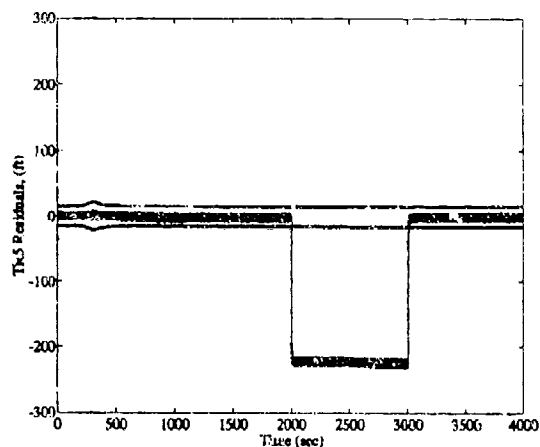
(Trans 2)



(Trans 3)

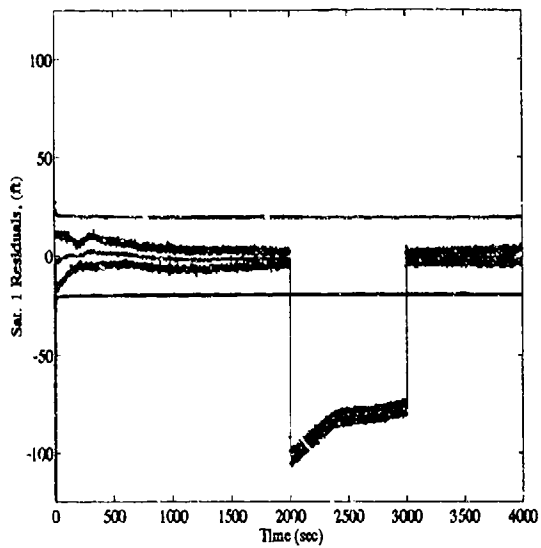


(Trans 4)

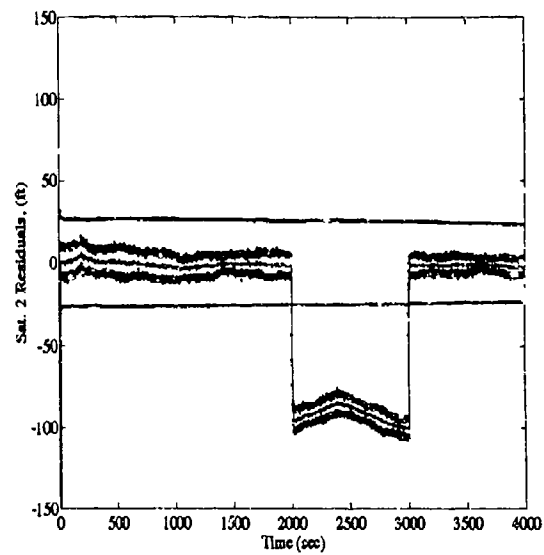


(Trans 5)

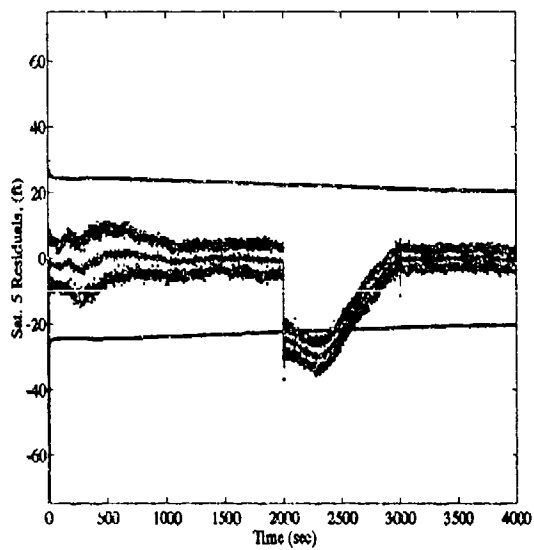
Figure E.4 NRS2 Transponder Scalar Residual Plots, Transponder Step Bias



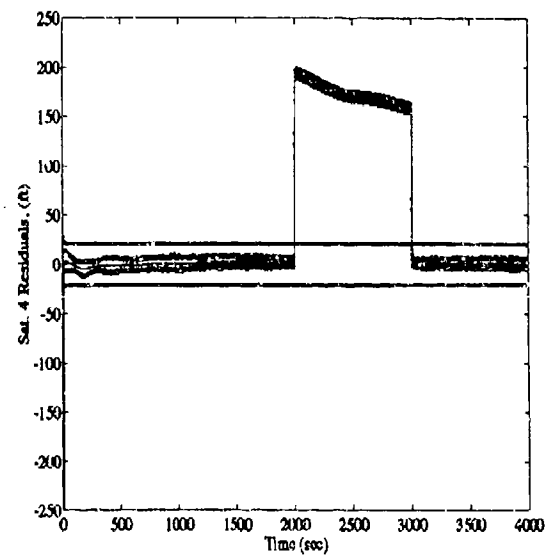
(Sat 1)



(Sat 2)

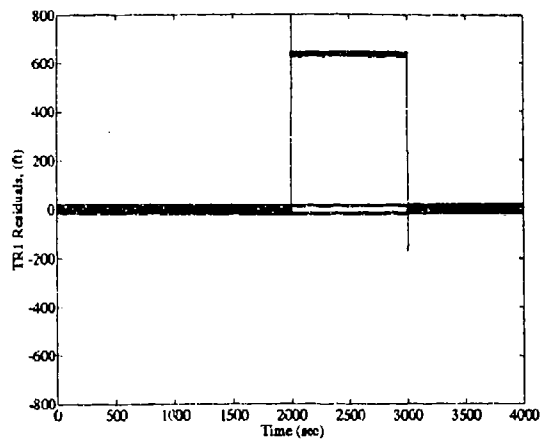


(Sat 5)

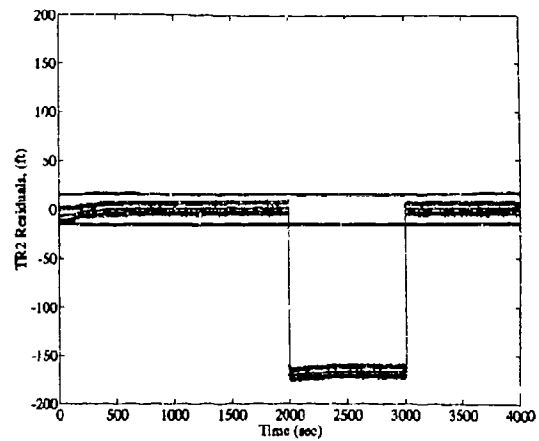


(Sat 4)

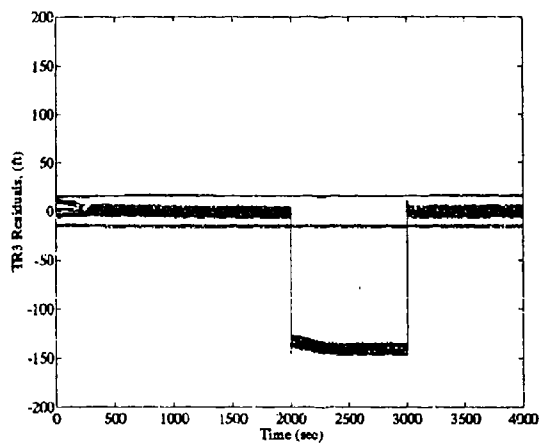
Figure E.5 NRS3 Satellite Scalar Residual Plots, Transponder Step Bias



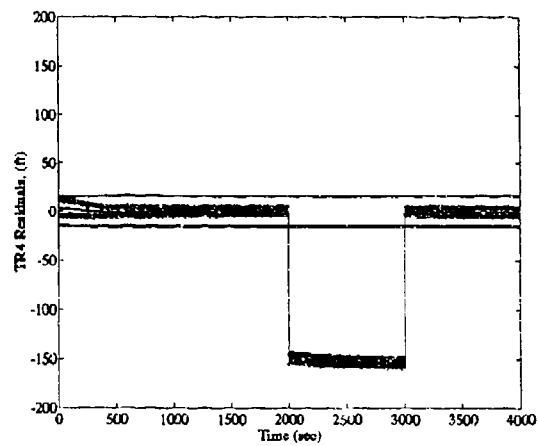
(Trans 1)



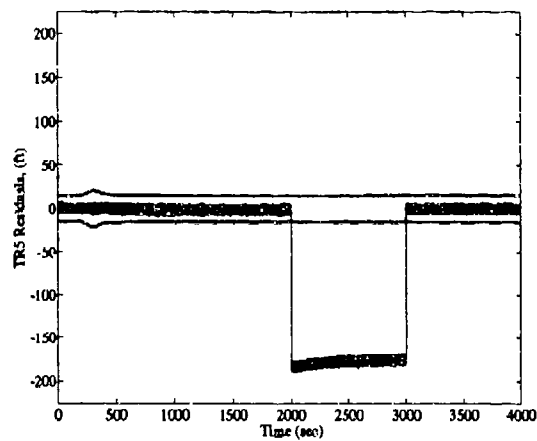
(Trans 2)



(Trans 3)

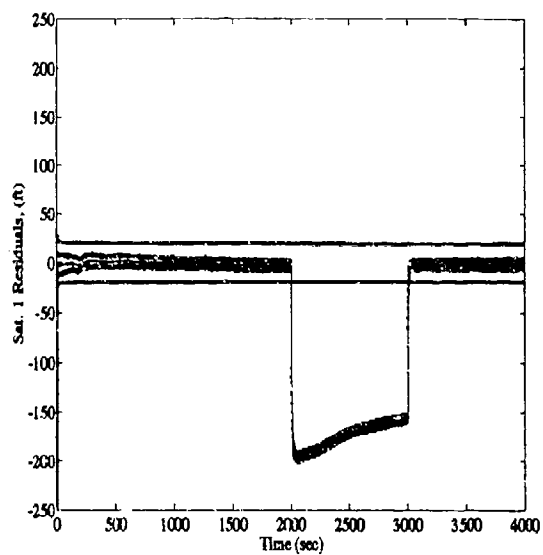


(Trans 4)

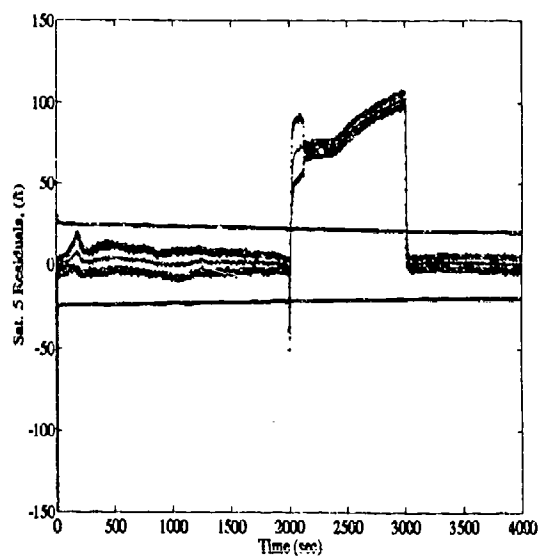


(Trans 5)

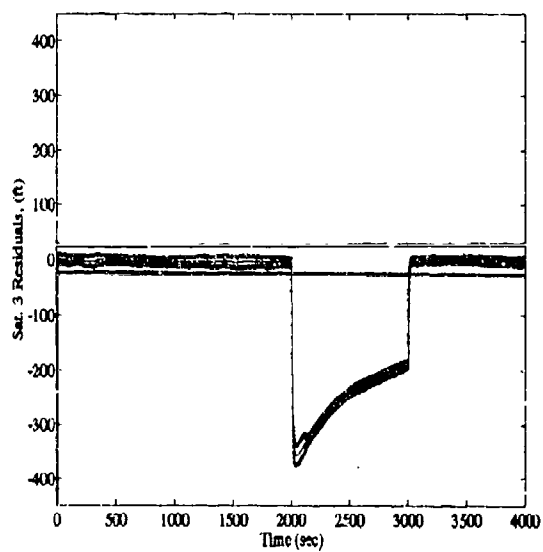
Figure E.6 NRS3 Transponder Scalar Residual Plots, Transponder Step Bias



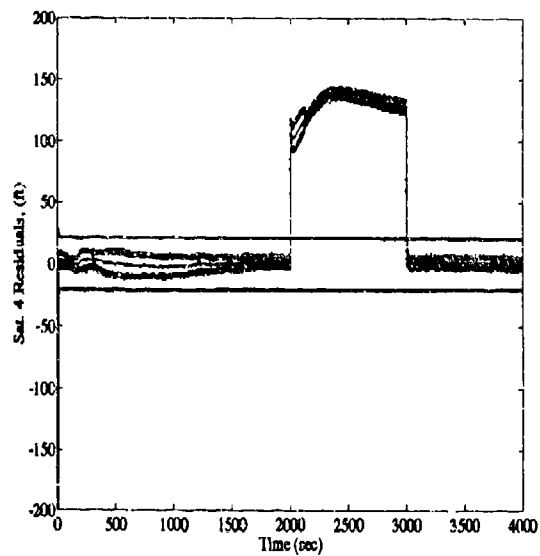
(Sat 1)



(Sat 5)

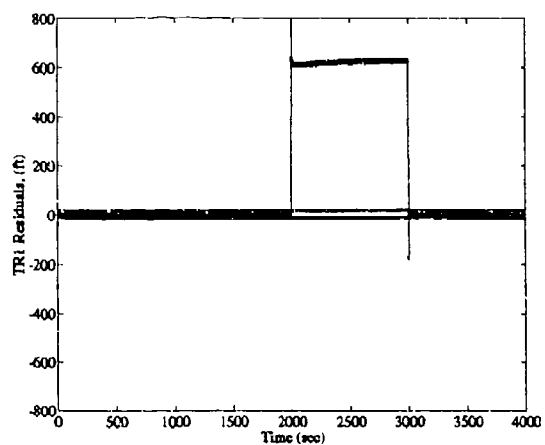


(Sat 3)

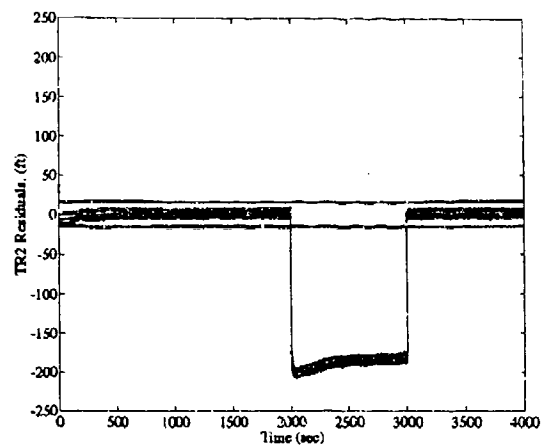


(Sat 4)

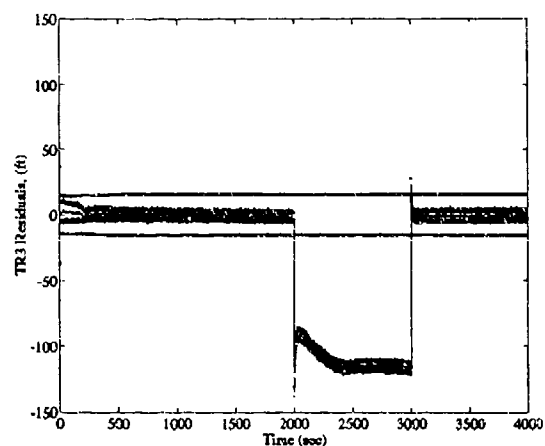
Figure E.7 NRS4 Satellite Scalar Residual Plots, Transponder Step Bias



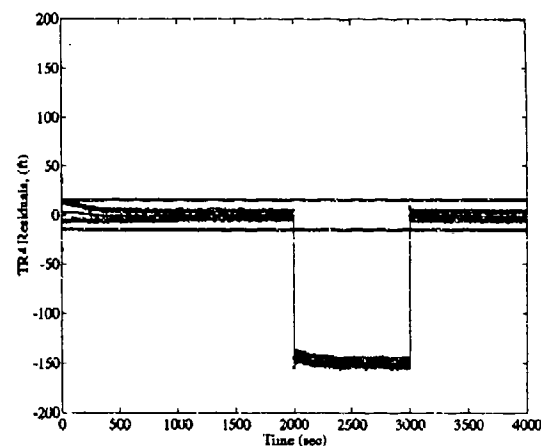
(Trans 1)



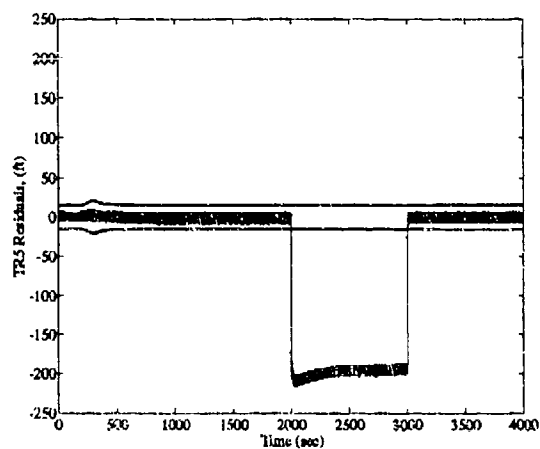
(Trans 2)



(Trans 3)

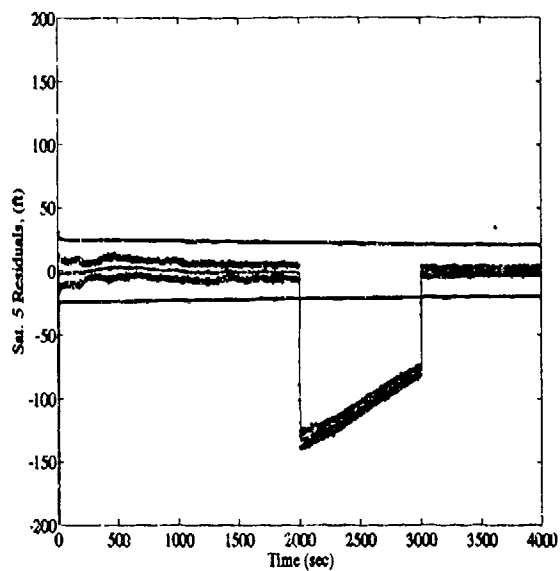


(Trans 4)

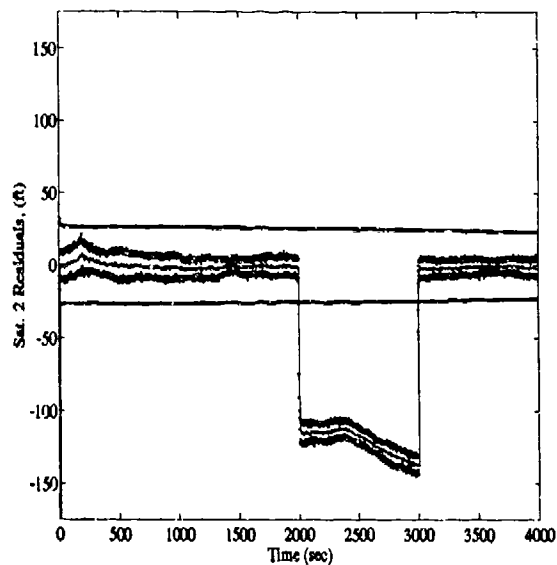


(Trans 5)

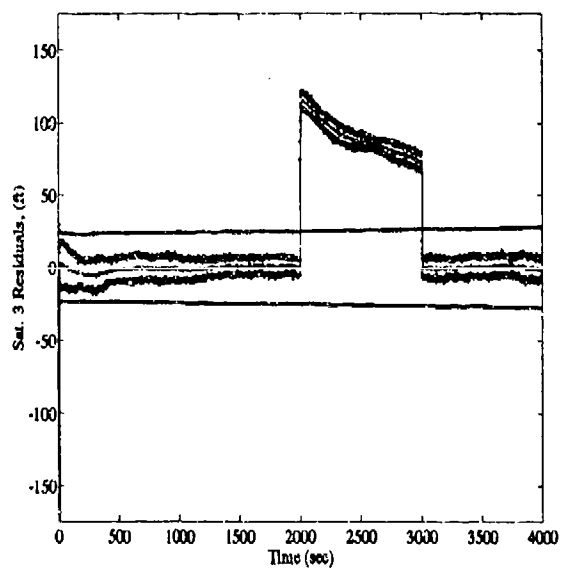
Figure E.8 NRS4 Transponder Scalar Residual Plots, Transponder Step Bias



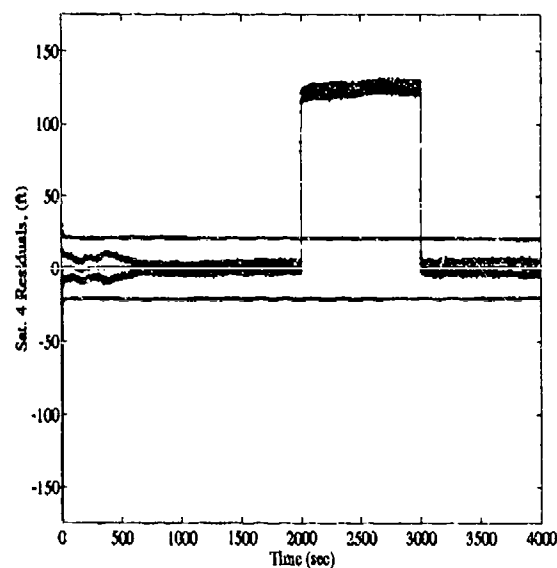
(Sat 5)



(Sat 2)

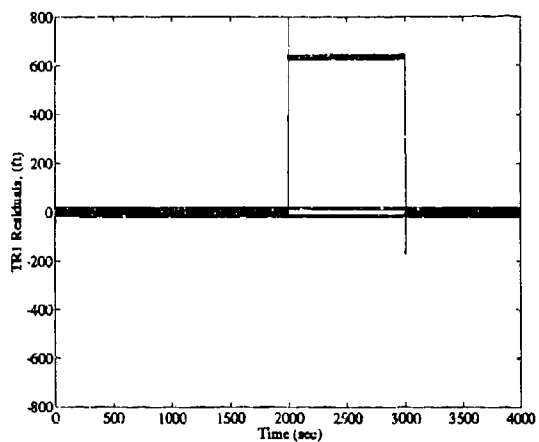


(Sat 3)

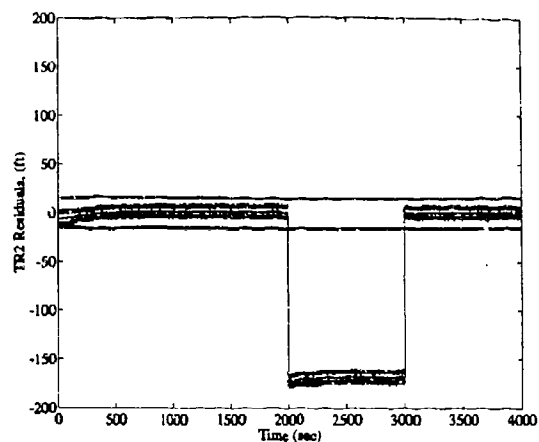


(Sat 4)

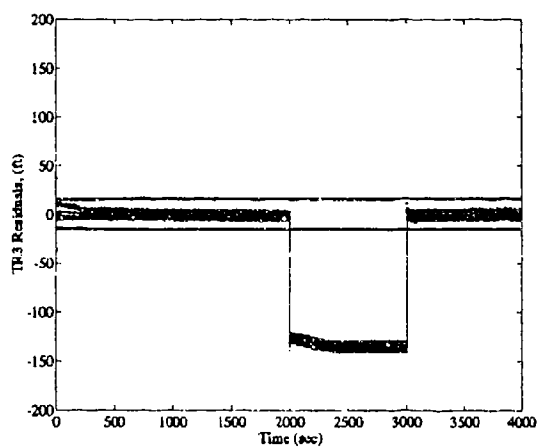
Figure E.9 NRS5 Satellite Scalar Residual Plots, Transponder Step Bias



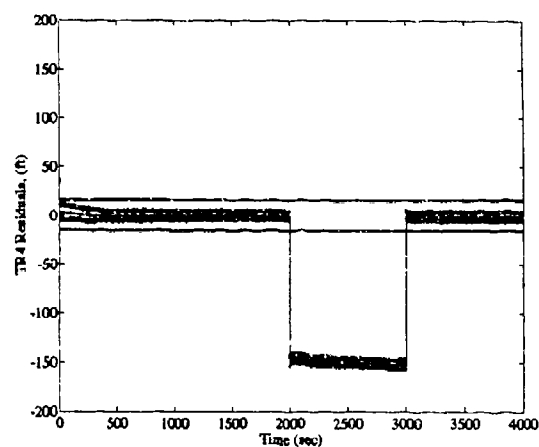
(Trans 1)



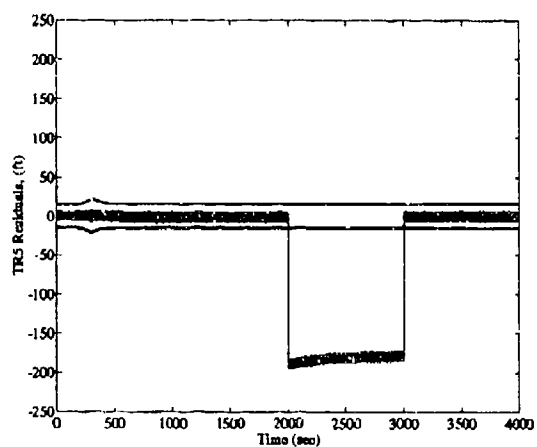
(Trans 2)



(Trans 3)

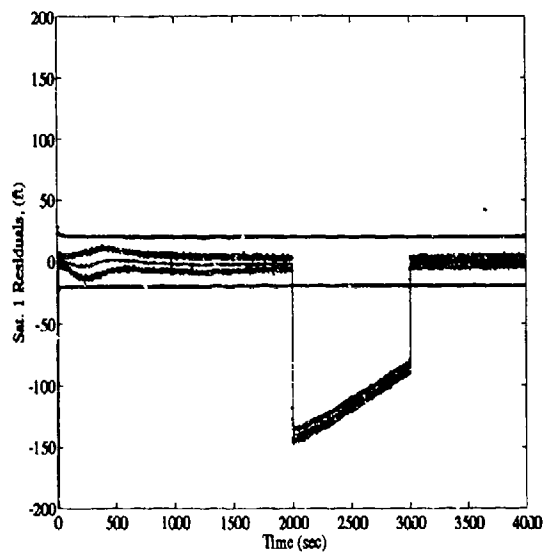


(Trans 4)

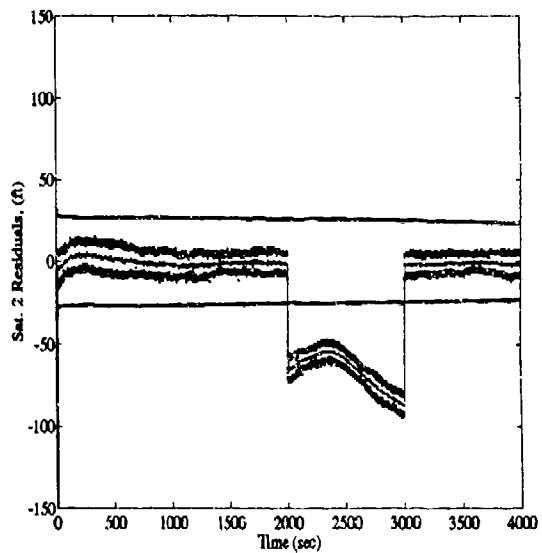


(Trans 5)

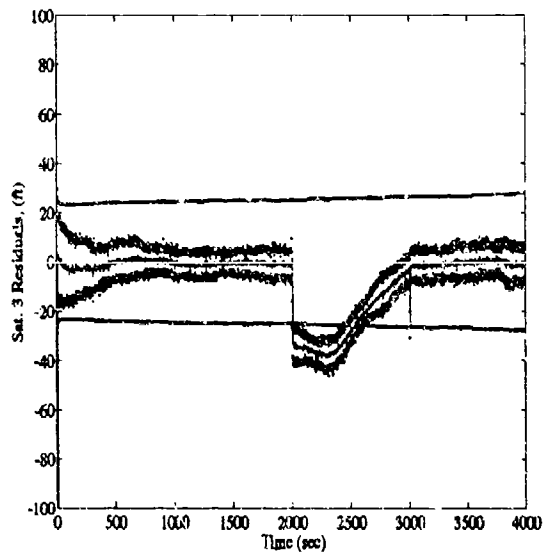
Figure E.10 NRS5 Transponder Scalar Residual Plots, Transponder Step Bias



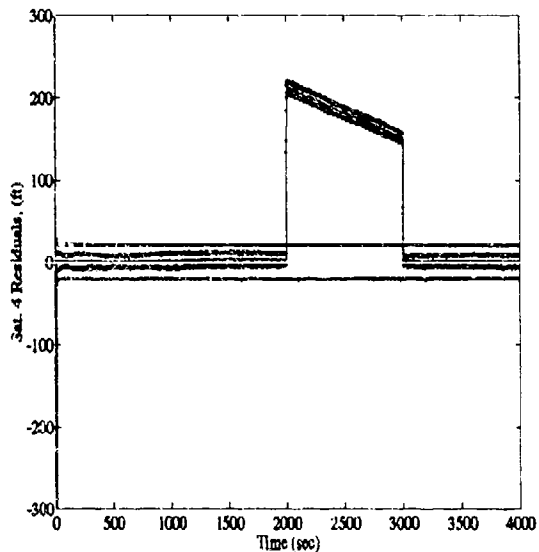
(Sat 1)



(Sat 2)

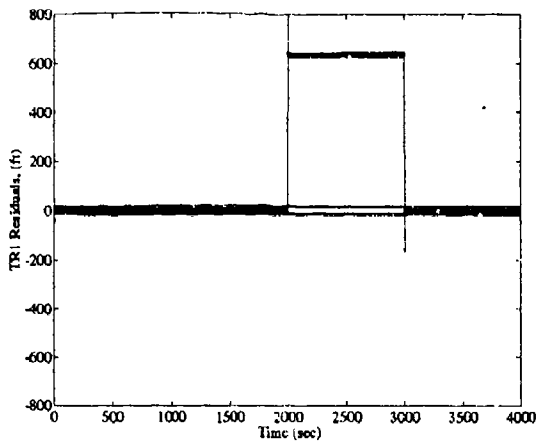


(Sat 3)

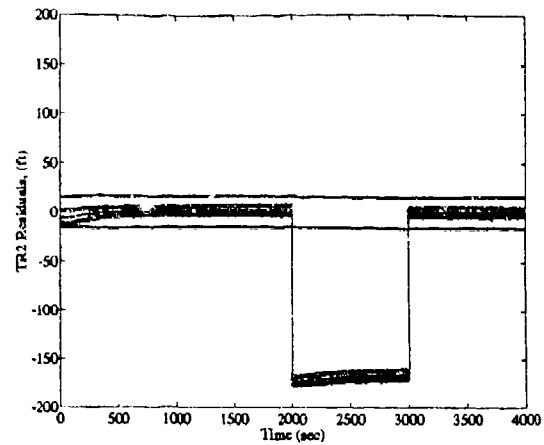


(Sat 4)

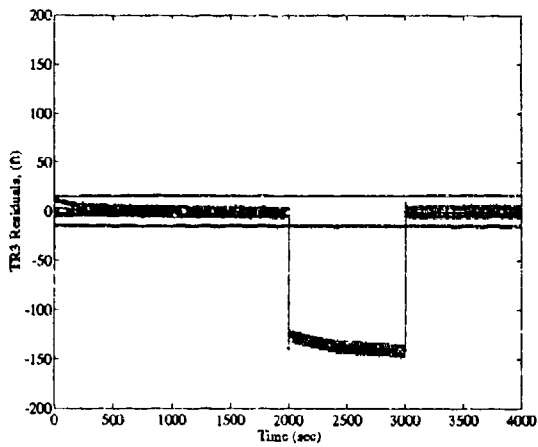
Figure E.11 NRS6 Satellite Scalar Residual Plots, Transponder Step Bias



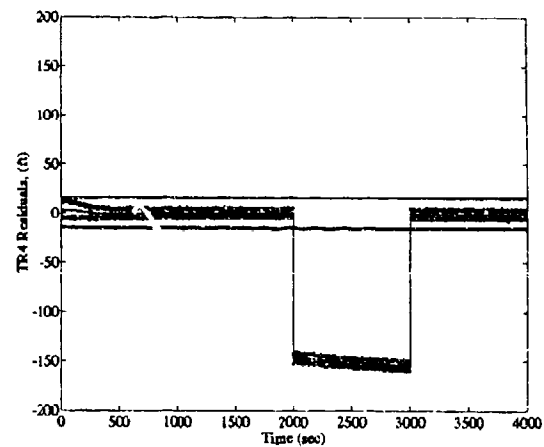
(Trans 1)



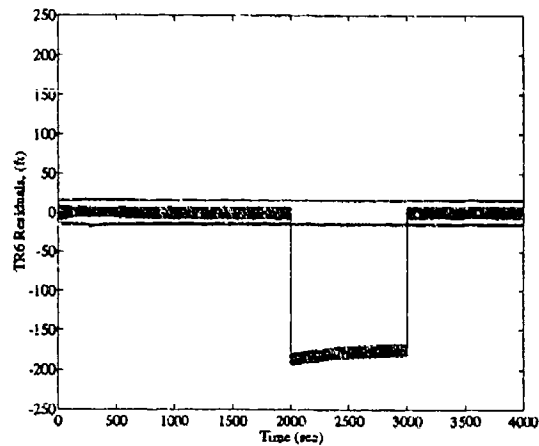
(Trans 2)



(Trans 3)

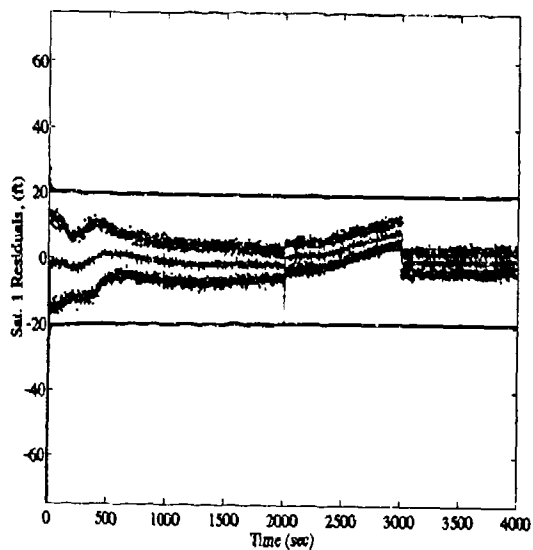


(Trans 4)

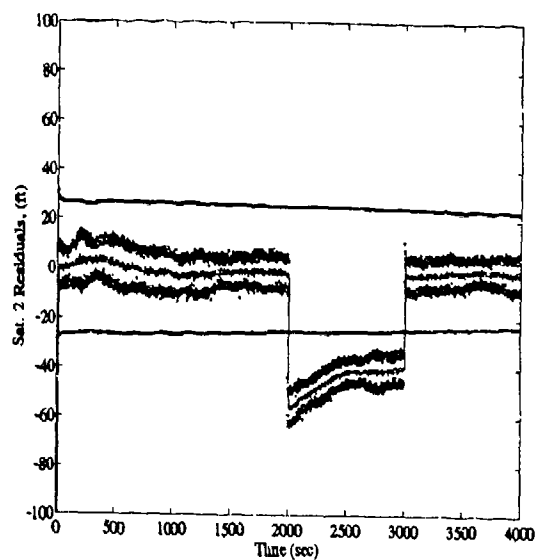


(Trans 6)

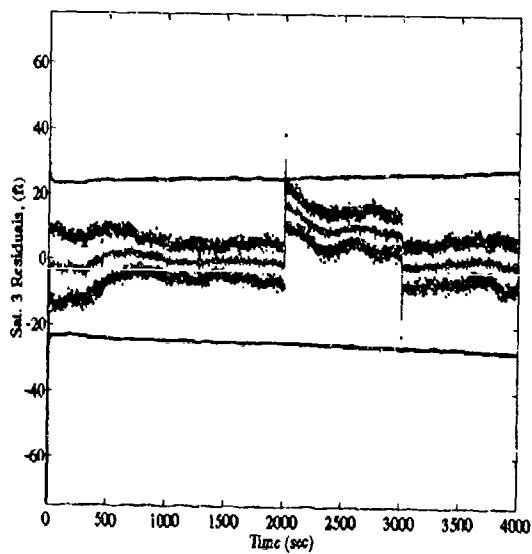
Figure E.12 NRS6 Transponder Scalar Residual Plots, Transponder Step Bias



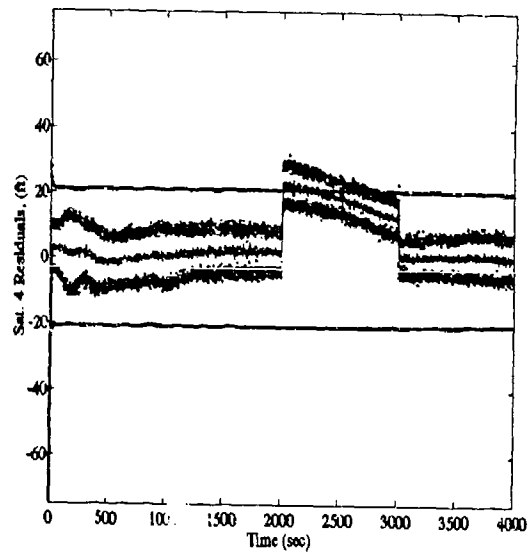
(Sat 1)



(Sat 2)

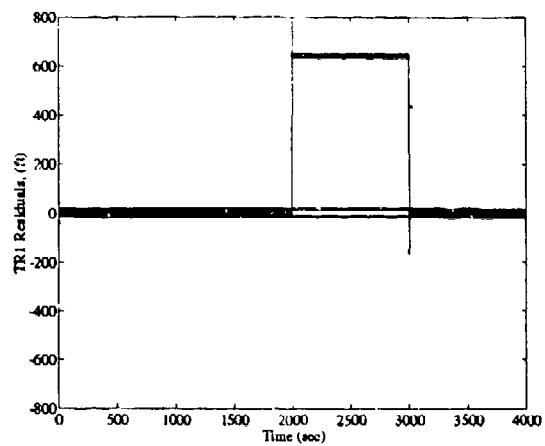


(Sat 3)

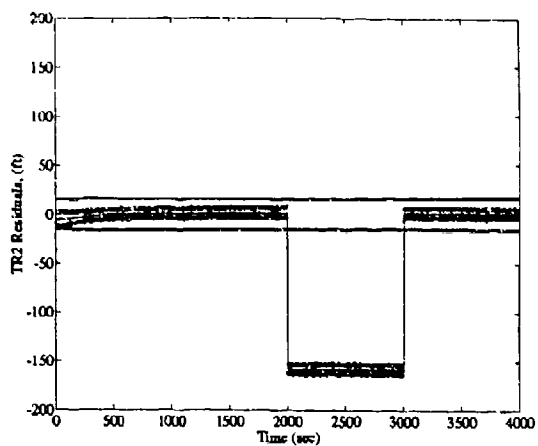


(Sat 4)

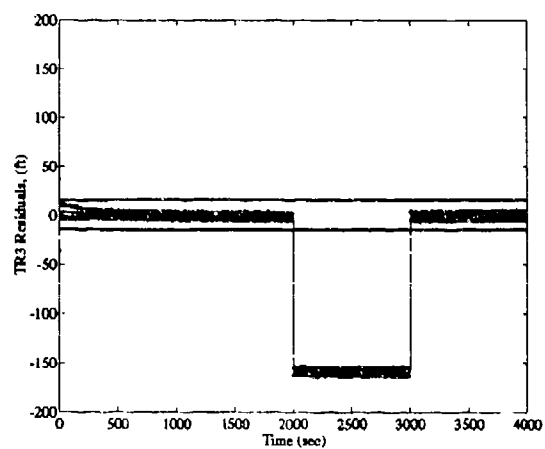
Figure E.13 NRS7 Satellite Scalar Residual Plots, Transponder Step Bias



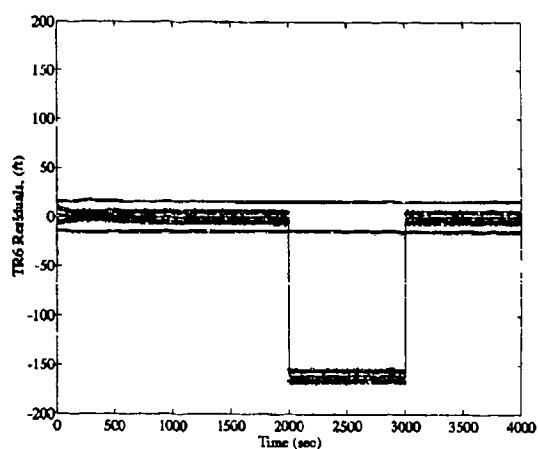
(Trans 1)



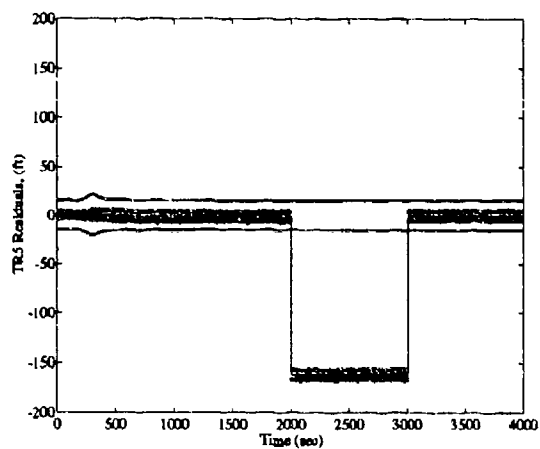
(Trans 2)



(Trans 3)

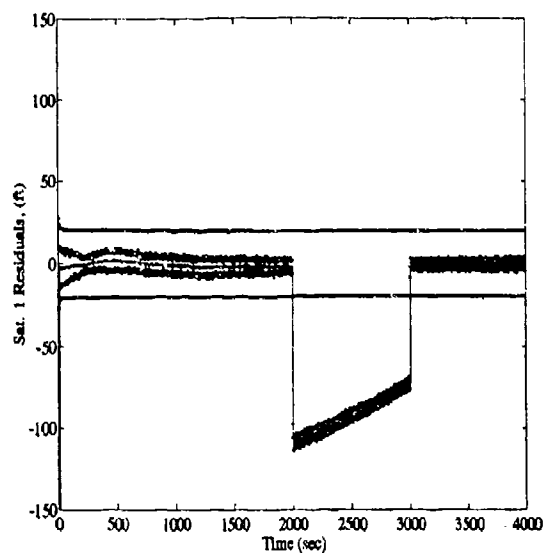


(Trans 6)

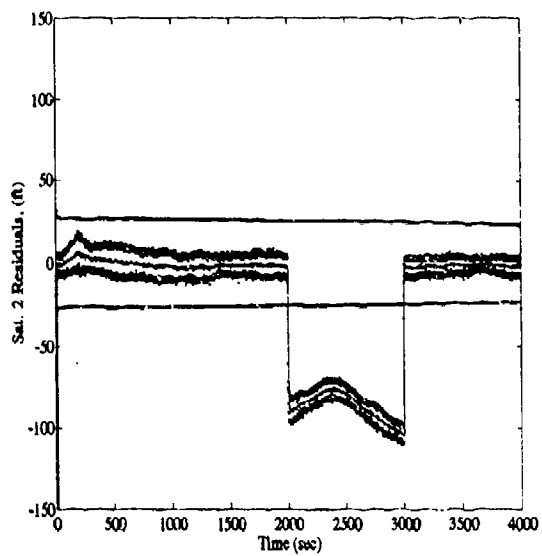


(Trans 5)

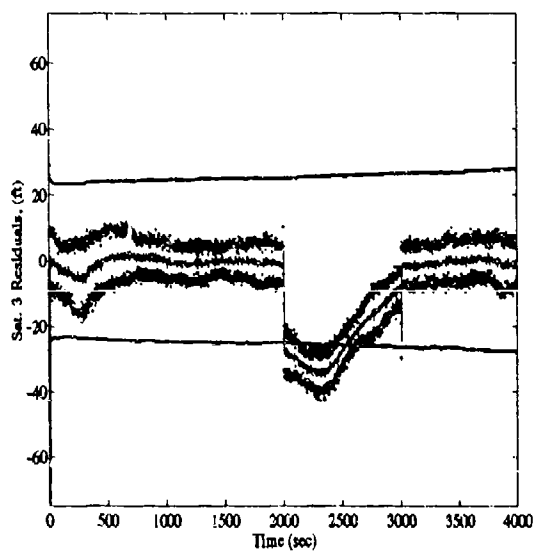
Figure E.14 NRS7 Transponder Scalar Residual Plots, Transponder Step Bias



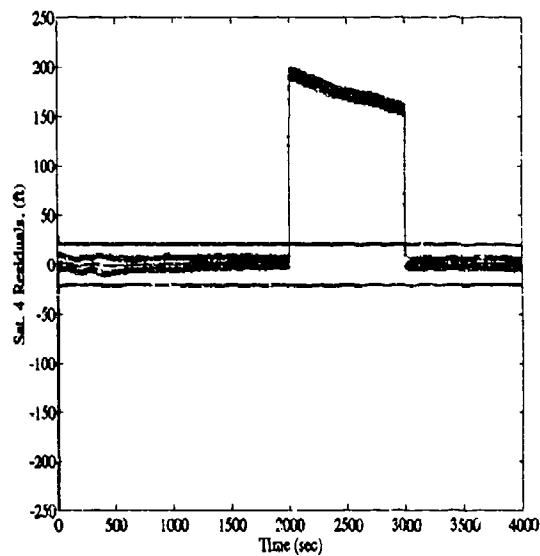
(Sat 1)



(Sat 2)

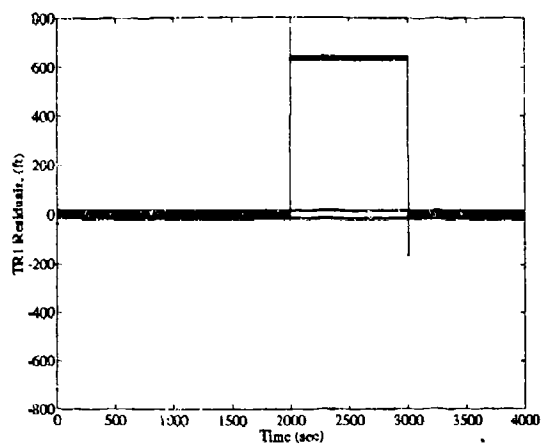


(Sat 3)

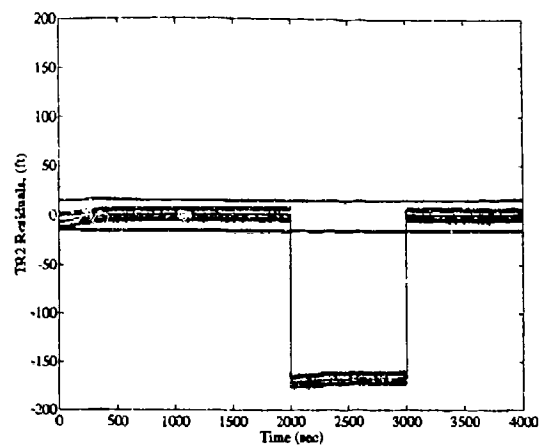


(Sat 4)

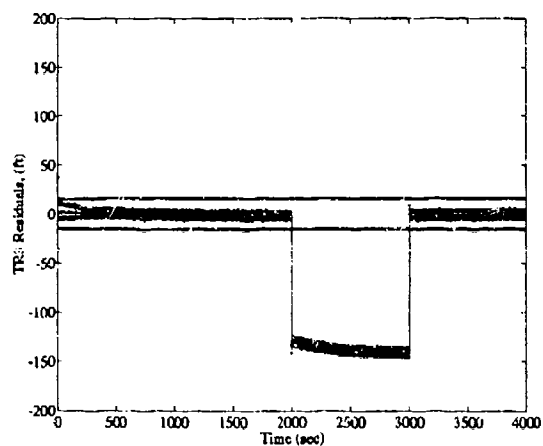
Figure E.15 NRS8 Satellite Scalar Residual Plots, Transponder Step Bias



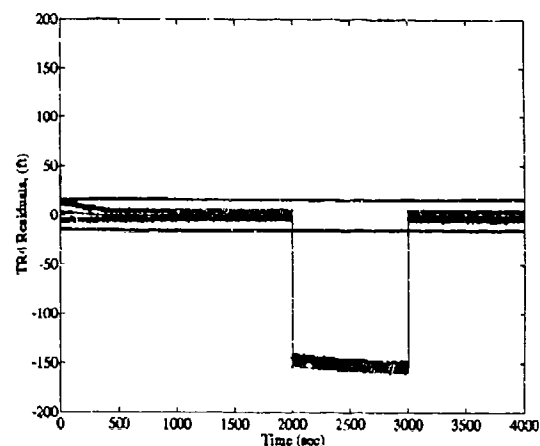
(Trans 1)



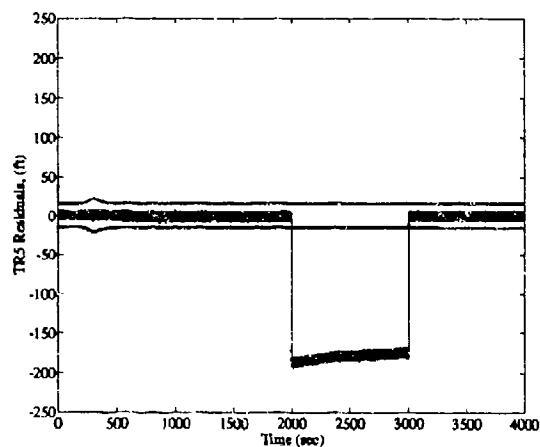
(Trans 2)



(Trans 6)

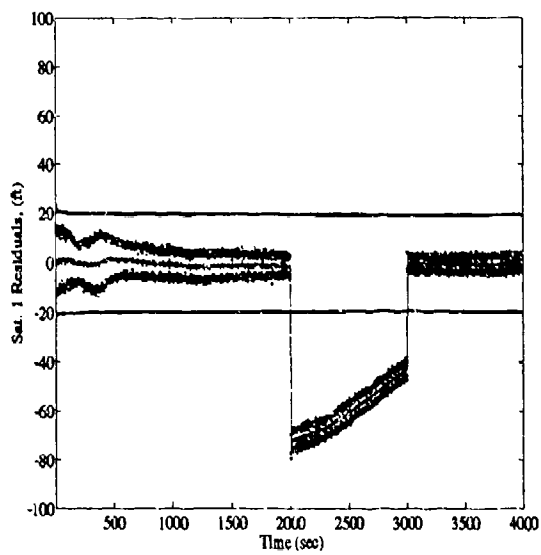


(Trans 4)

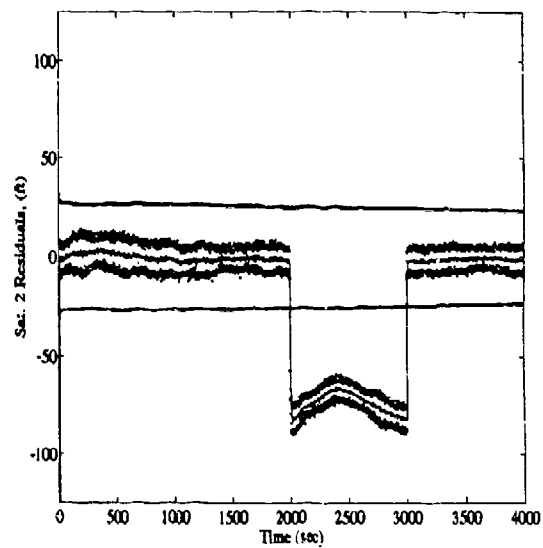


(Trans 5)

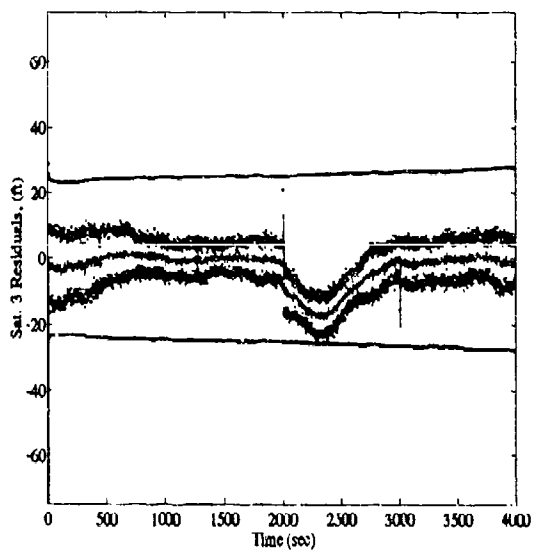
Figure E.16 NRS8 Transponder Scalar Residual Plots, Transponder Step Bias



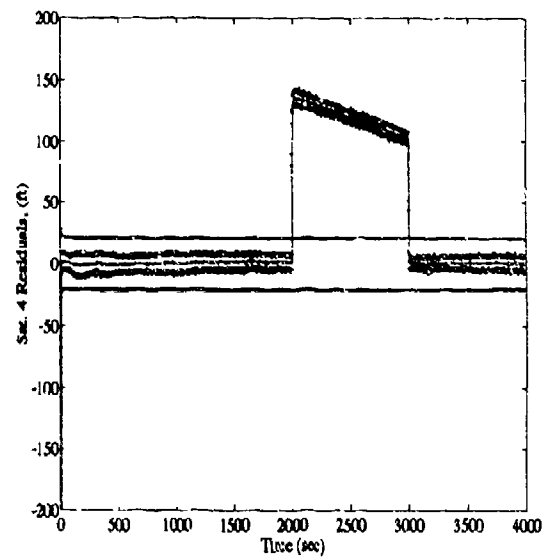
(Sat 1)



(Sat 2)

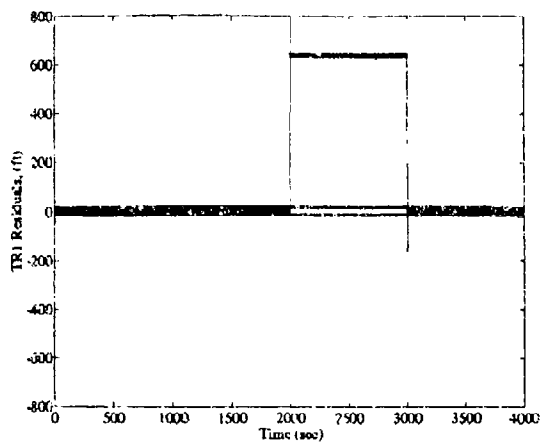


(Sat 3)

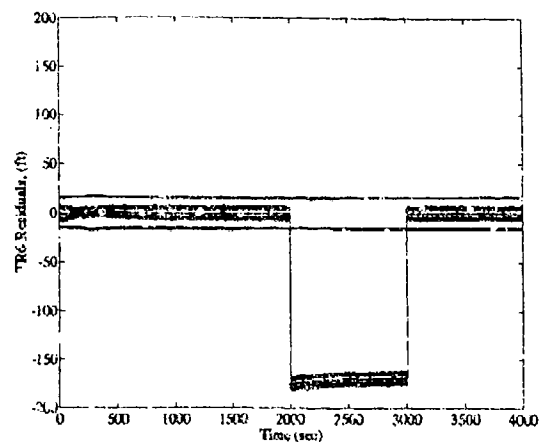


(Sat 4)

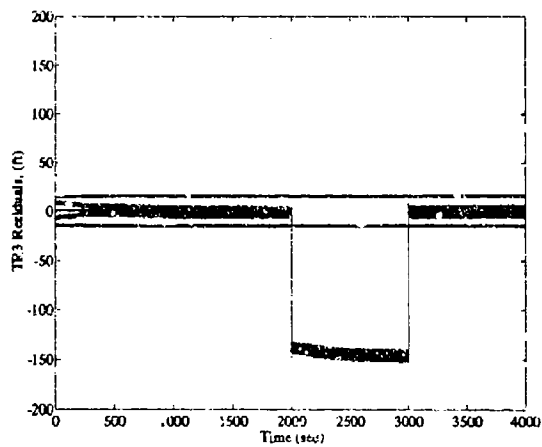
Figure E.17 NRS9 Satellite Scalar Residual Plots, Transponder Step Bias



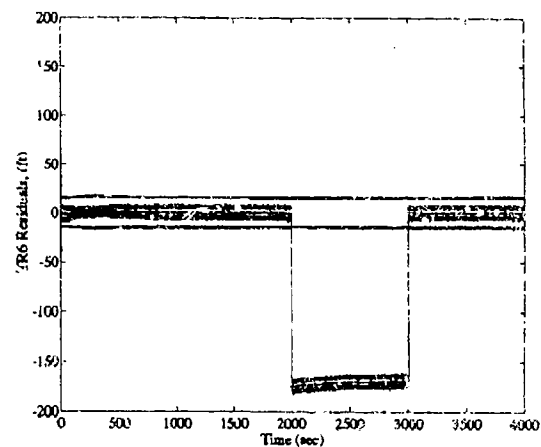
(Trans 1)



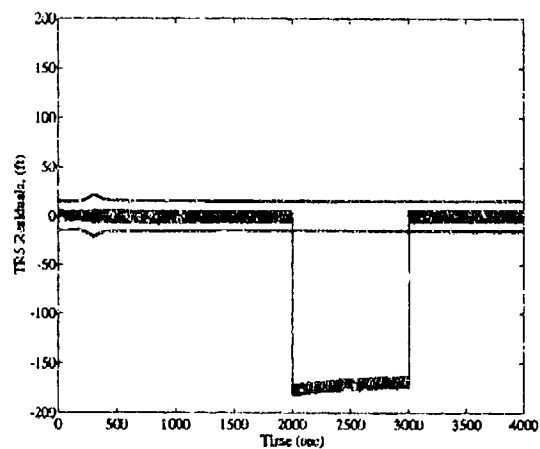
(Trans 6)



(Trans 3)

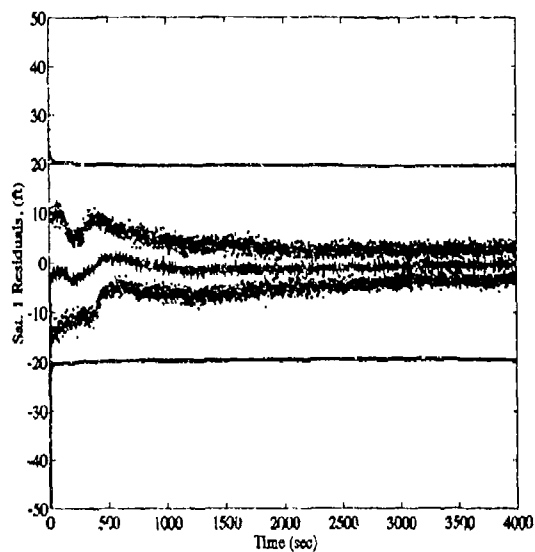


(Trans 4)

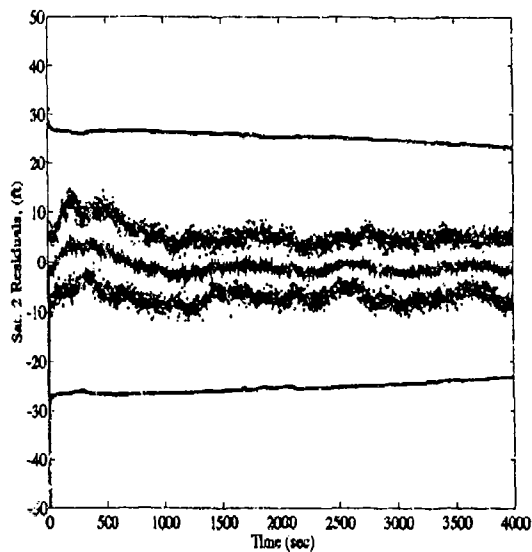


(Trans 5)

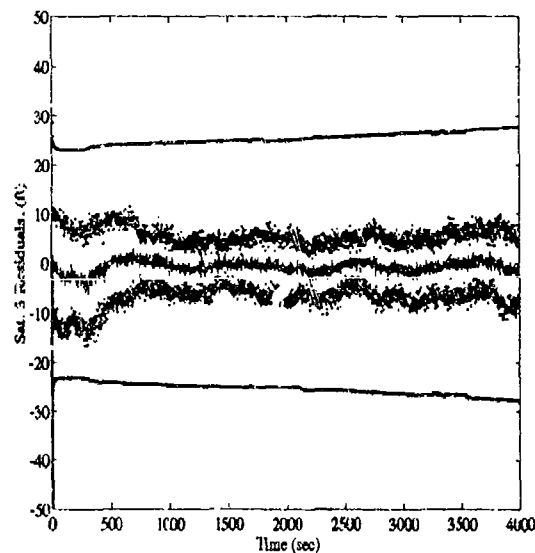
Figure E.18 NRS9 Transponder Scalar Residual Plots, Transponder Step Bias



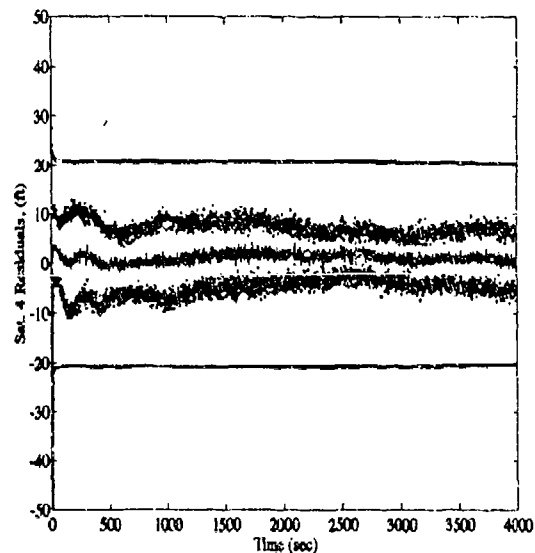
(Sat 1)



(Sat 2)

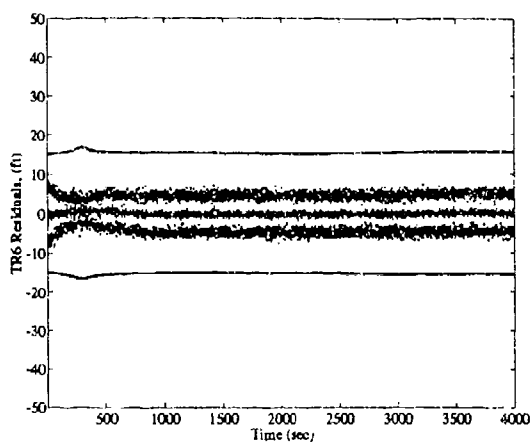


(Sat 3)

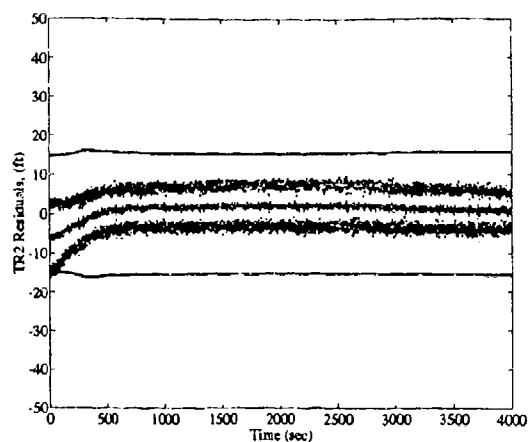


(Sat 4)

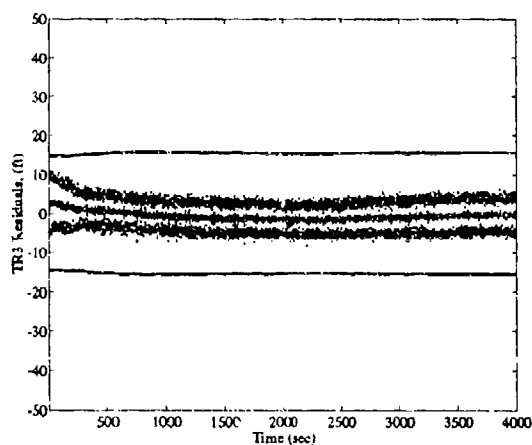
Figure E.19 NRS10 Satellite Scalar Residual Plots, Transponder Step Bias



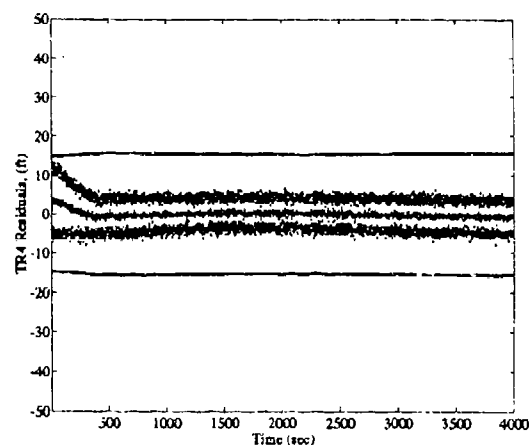
(Trans 6)



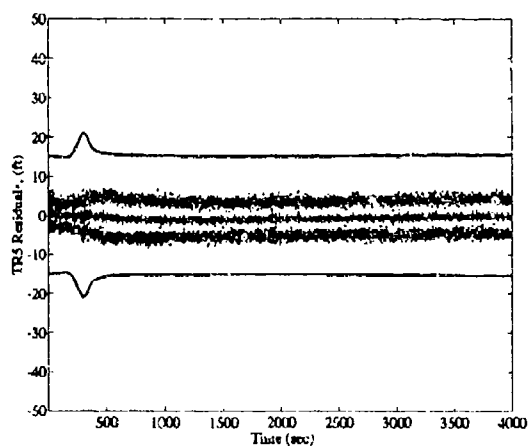
(Trans 2)



(Trans 3)



(Trans 4)



(Trans 5)

Figure E.20 NRS10 Transponder Scalar Residual Plots, Transponder Step Bias

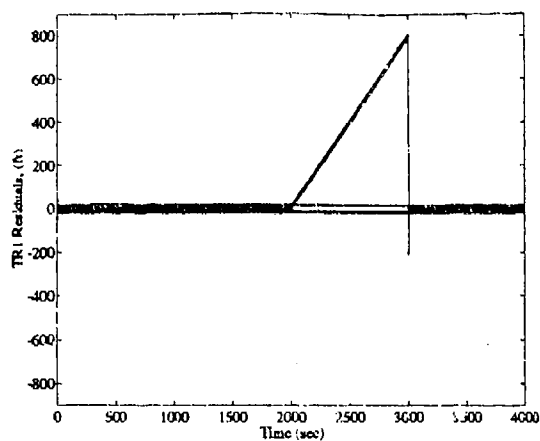
Appendix F. Residual Plots for a Transponder Signal Ramp Failure

This appendix contains limited scalar residual data for the ten NRS filters during a simulated ramp offset of slope $[1(t - 2000)]\text{ft}$ on the Transponder 1 range signal from time $t = 2000\text{sec}$ through $t = 3000\text{sec}$. The plots contained in this section are identical in format to those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

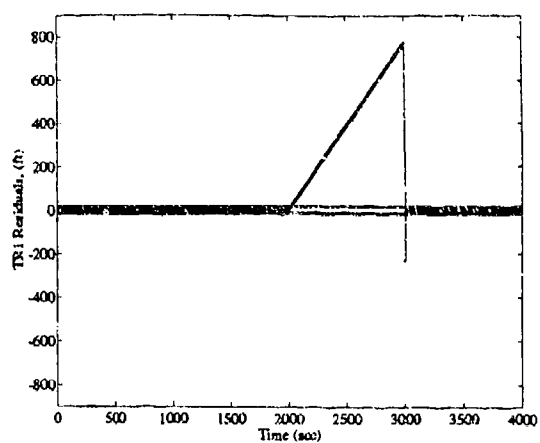
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. Unlike Appendices D, E, and H, this section contains only the scalar residual plots for the failed measurement signal (Transponder 1) in nine of the ten filters. Filter 10 is not included, because it does not receive the failed measurement. The first plot presented is for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS9. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table F.1 Legend for Filter Tuning Plots

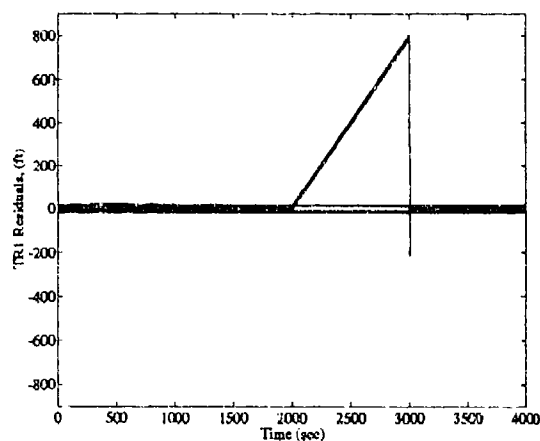
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
-- Dashed Line	\pm Filter-Predicted Sigma



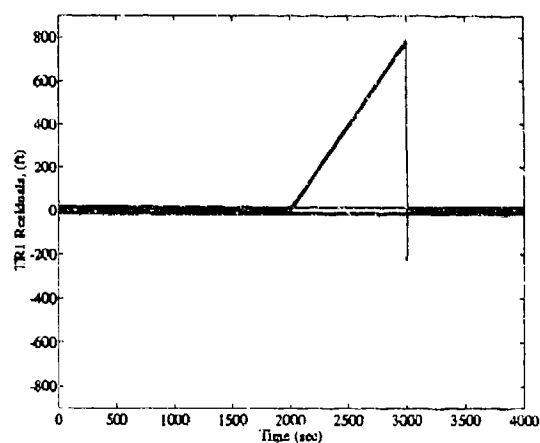
(Filter 1)



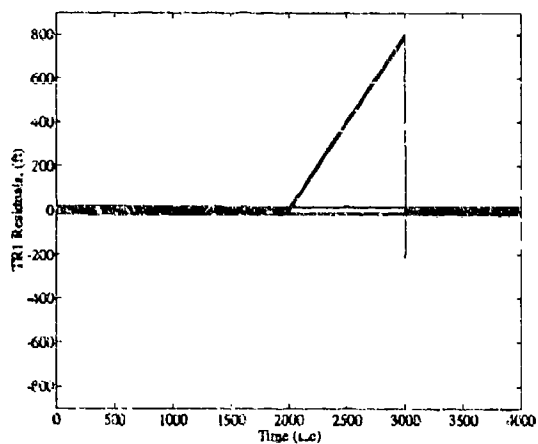
(Filter 2)



(Filter 3)

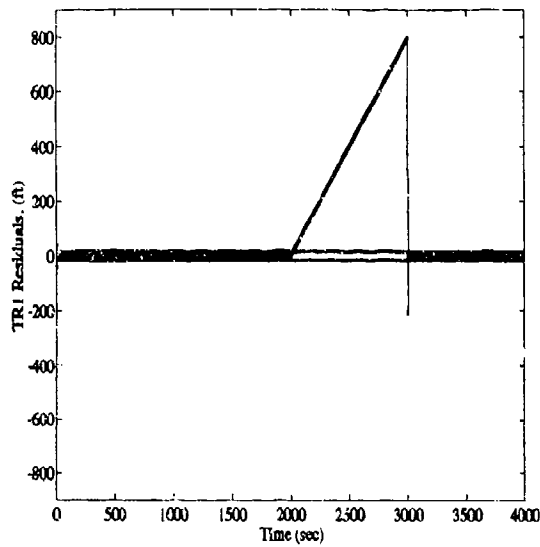


(Filter 4)

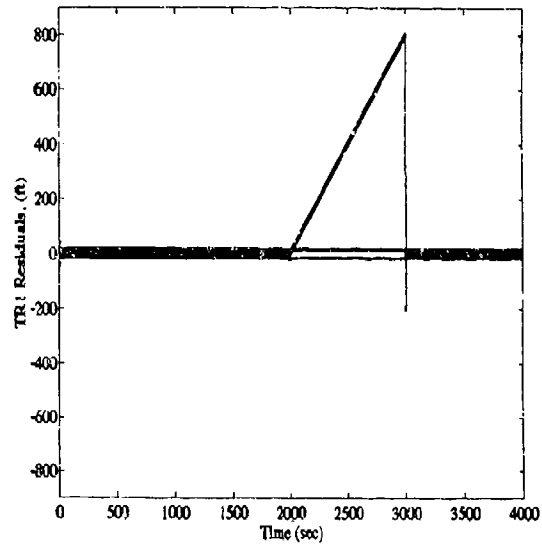


(Filter 5)

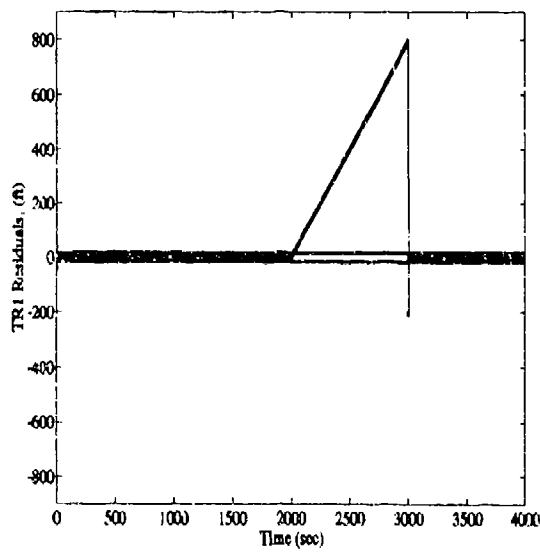
Figure F.1 Transponder 1 Scalar Residual Plots, Transponder Ramp Increase for NRS Filters 1-5



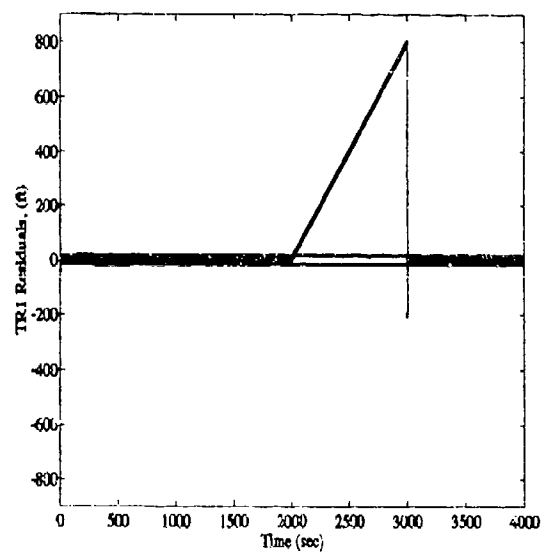
(Filter 6)



(Filter 7)



(Filter 8)



(Filter 9)

Figure F.2 Transponder 1 Scalar Residual Plots, Transponder Ramp Increase for NRS Filters 6-9

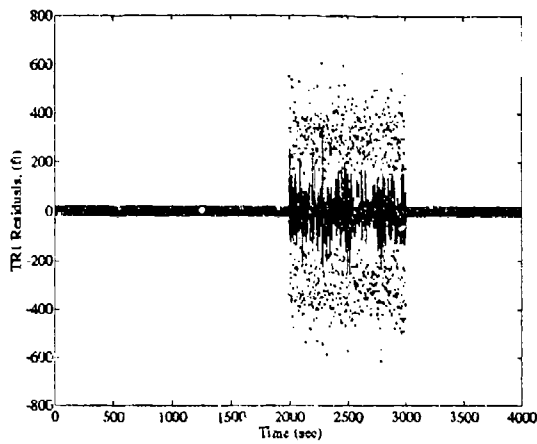
Appendix G. Residual Plots for a Transponder Signal Noise Failure

This appendix contains limited scalar residual data for the ten NRS filters during a simulated noise increase of (150v)ft on the Transponder 1 range signal from time $t = 2000sec$ through $t = 3000sec$. The plots contained in this section are identical in format to those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

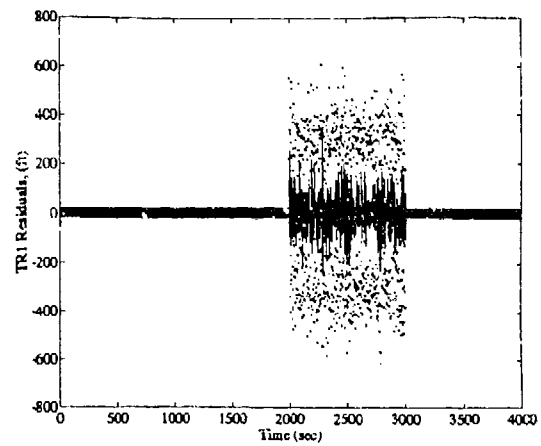
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. Unlike Appendices D, E, and H, this section contains only the scalar residual plots for the failed measurement signal (Transponder 1) in nine of the ten filters. Filter 10 is not included, because it does not receive the failed measurement. The first plot presented is for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS9. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table G.1 Legend for Filter Tuning Plots

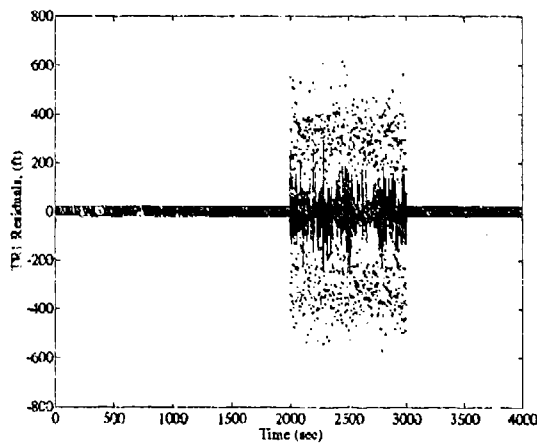
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
-- Dashed Line	\pm Filter-Predicted Sigma



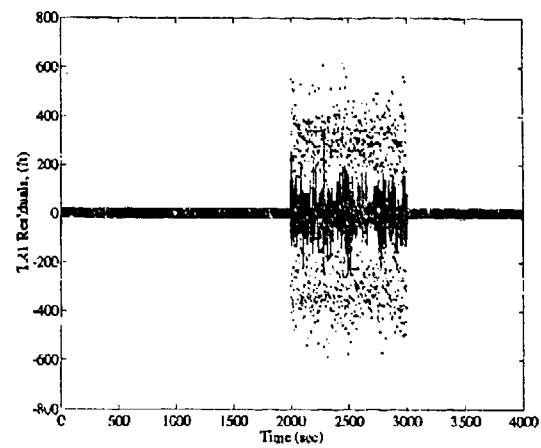
(Filter 1)



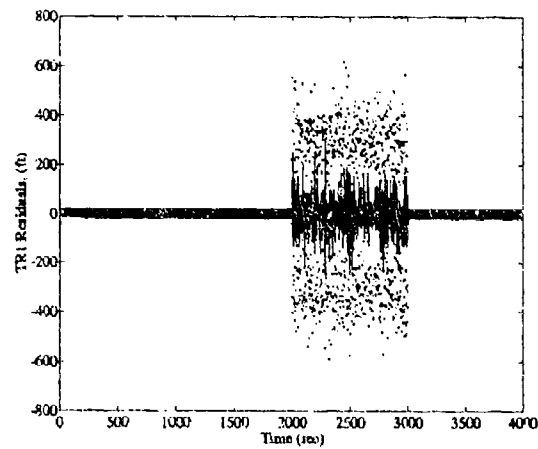
(Filter 2)



(Filter 3)

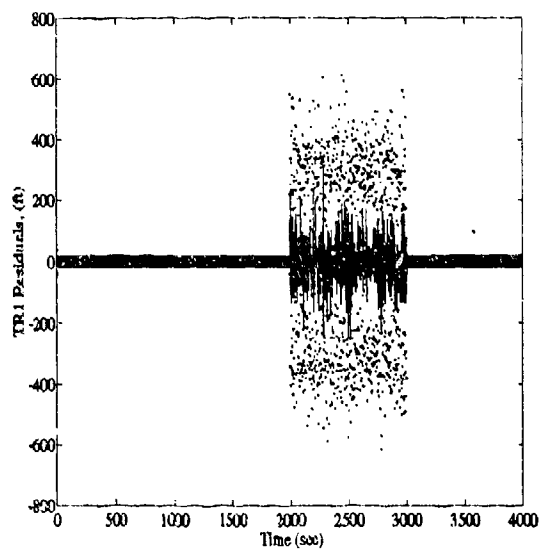


(Filter 4)

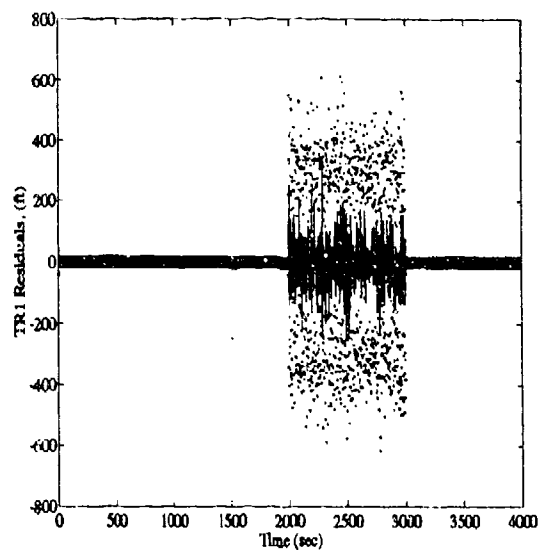


(Filter 5)

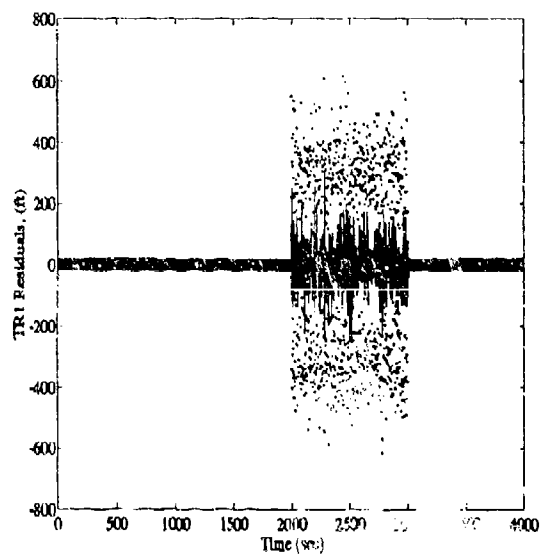
Figure G.1 Transponder 1 Scalar Residual Plots, Transponder Noise Increase for NRS Filters 1-5



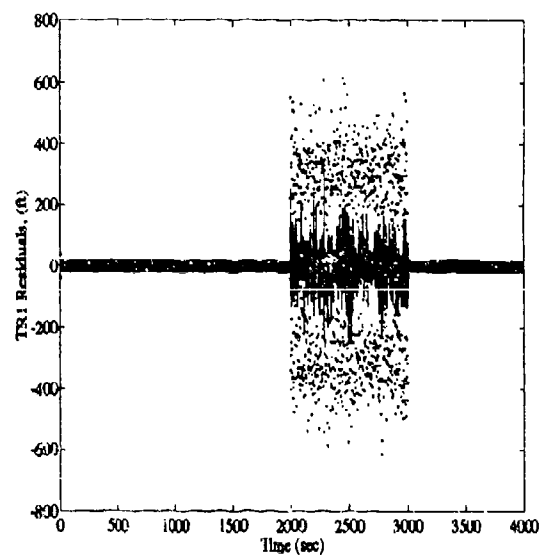
(Filter 6)



(Filter 7)



(Filter 8)



(Filter 9)

Figure G.2 Transponder 1 Scalar Residual Plots, Transponder Noise Increase for NRS Filters 6-9

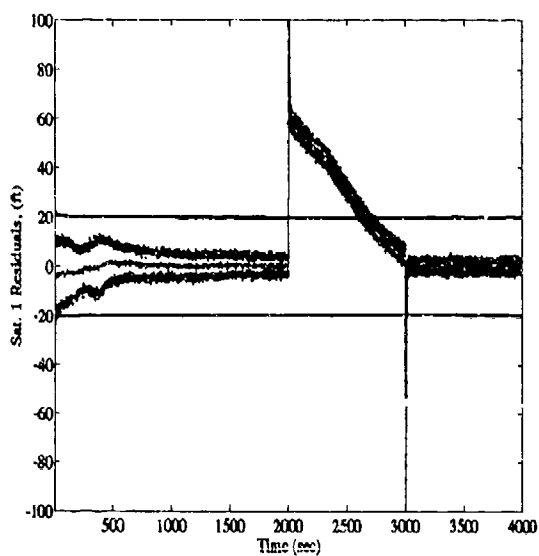
Appendix H. Residual Plots for a Satellite Signal Step Failure

This appendix contains limited scalar residual data for the ten NRS filters during a simulated step bias of 3500ft on the Satellite 3 pseudorange signal from time $t = 2000\text{sec}$ through $t = 3000\text{sec}$. The plots contained in this section are identical in format to those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

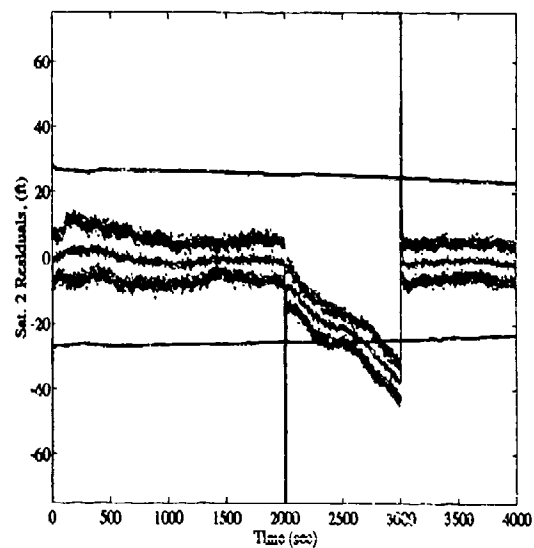
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. This section contains the plots for the nine measurement updates (4 Satellites and 5 Transponders) in each of the *ten* NRS filters. The first plots presented are for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS10. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table H.1 Legend for Filter Tuning Plots

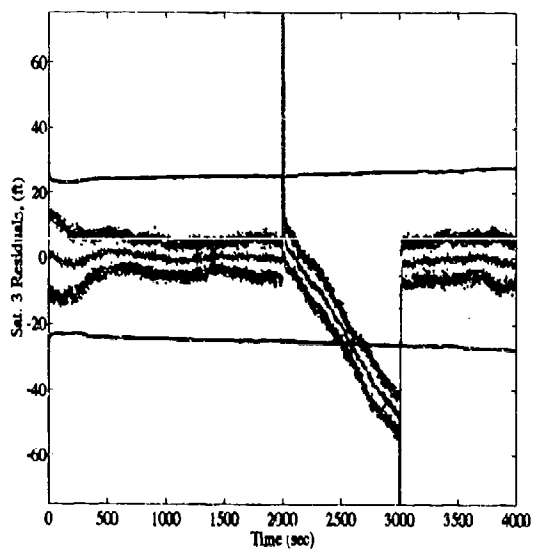
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
-- Dashed Line	\pm Filter-Predicted Sigma



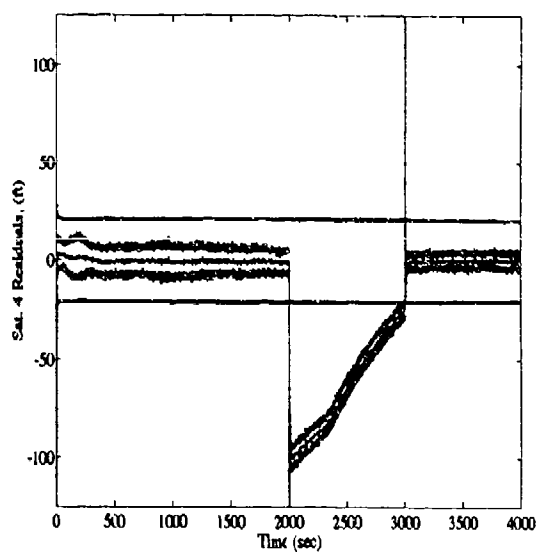
(Sat 1)



(Sat 2)

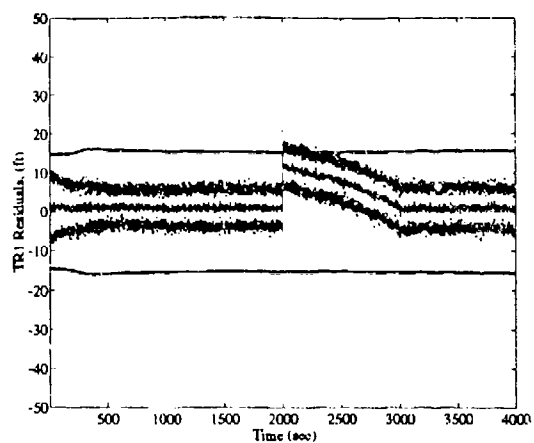


(Sat 3)

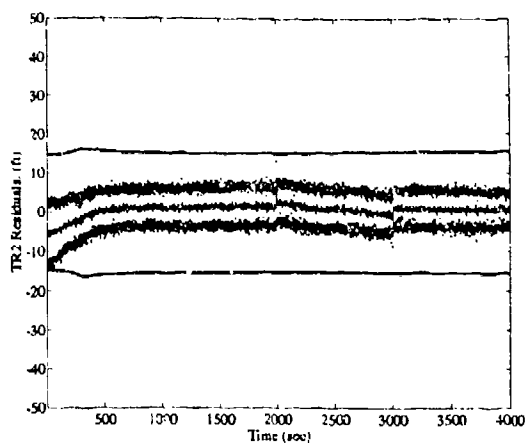


(Sat 4)

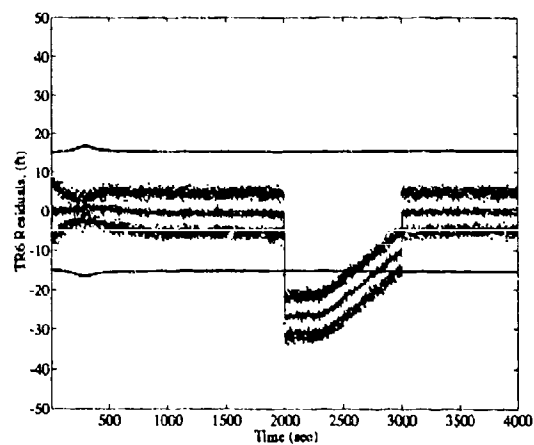
Figure H.1 NRS1 Satellite Scalar Residual Plots, Satellite Step Bias



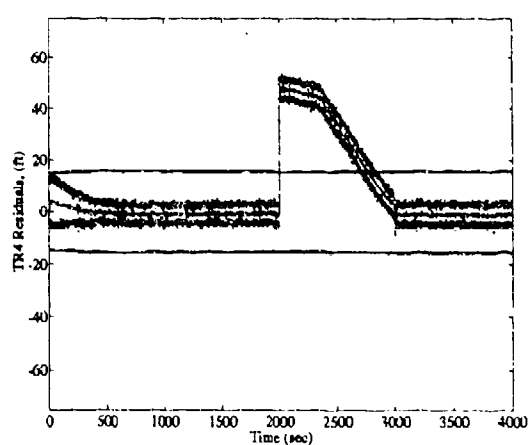
(Trans 1)



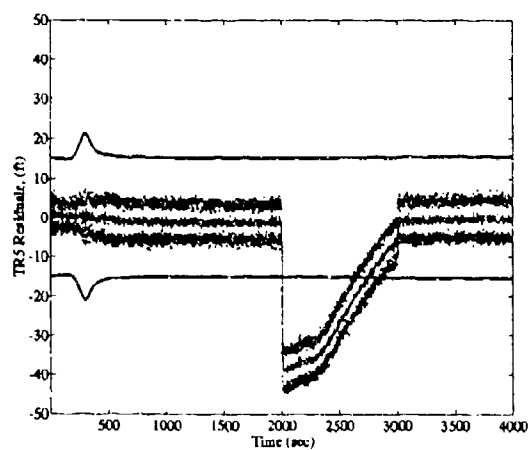
(Trans 2)



(Trans 3)

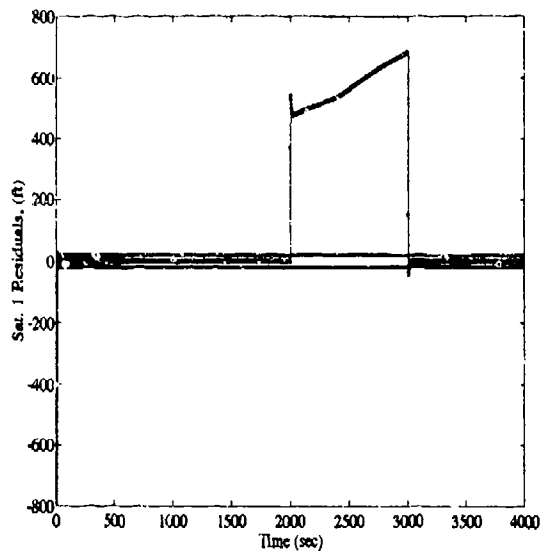


(Trans 4)

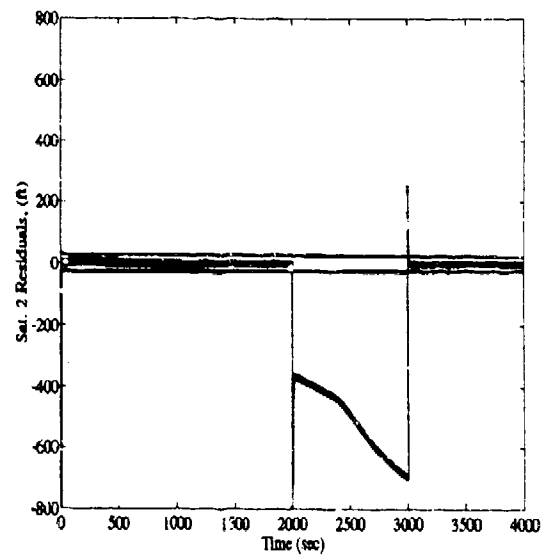


(Trans 5)

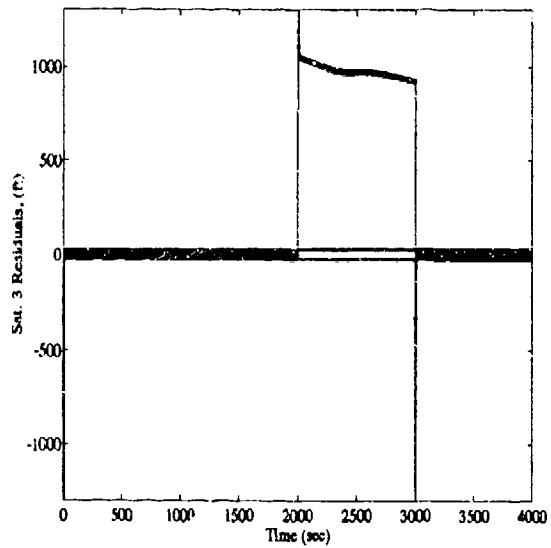
Figure H.2 NRS1 Transponder Scalar Residual Plots, Satellite Step Bias



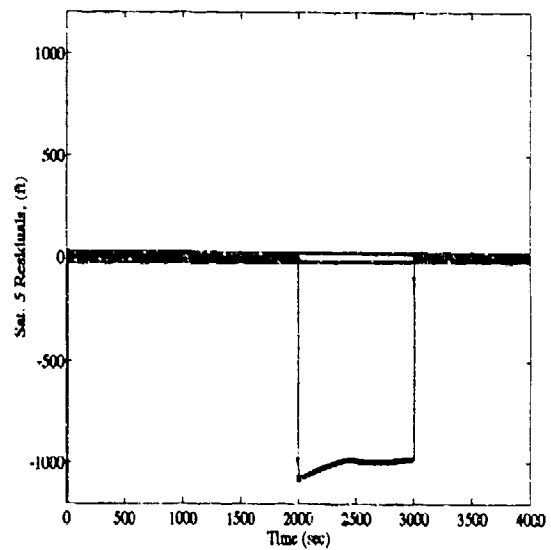
(Sat 1)



(Sat 2)

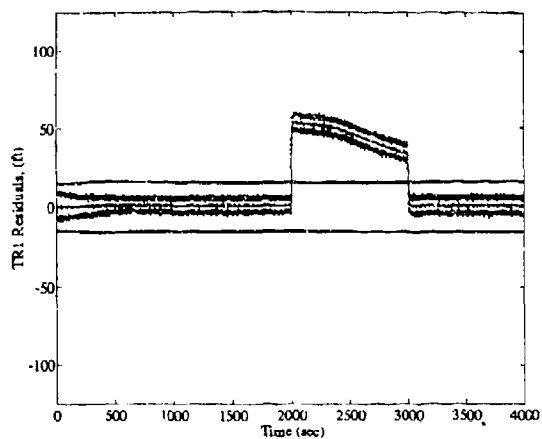


(Sat 3)

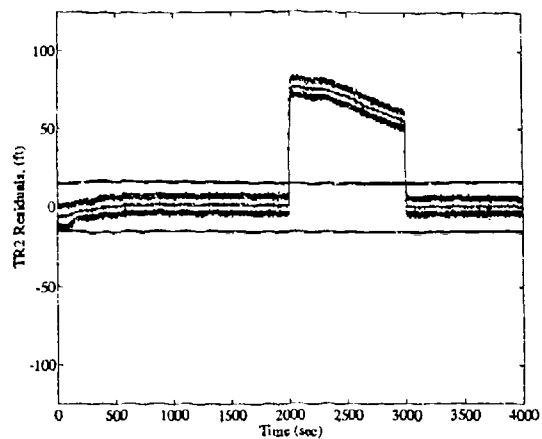


(Sat 5)

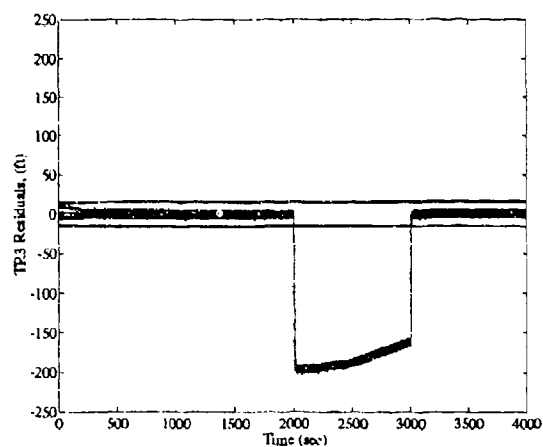
Figure H.3 NRS2 Satellite Scalar Residual Plots, Satellite Step Bias



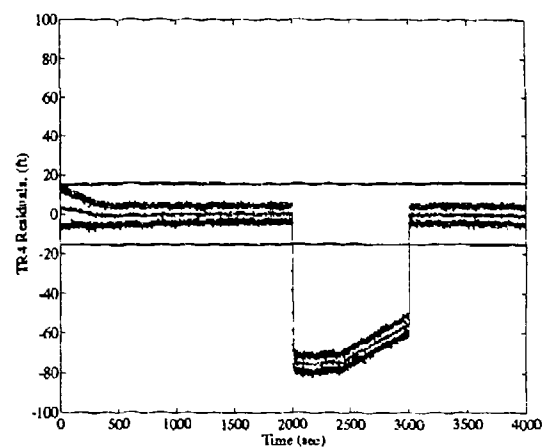
(Trans 1)



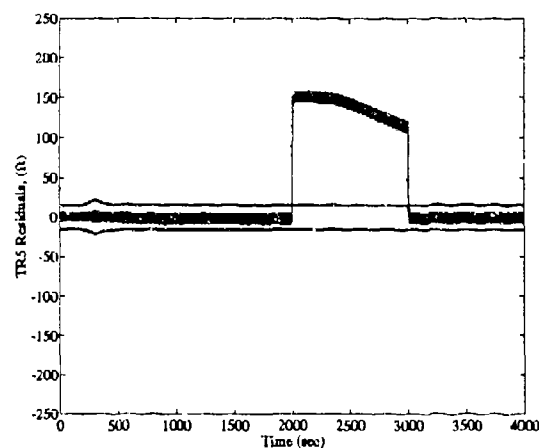
(Trans 2)



(Trans 3)

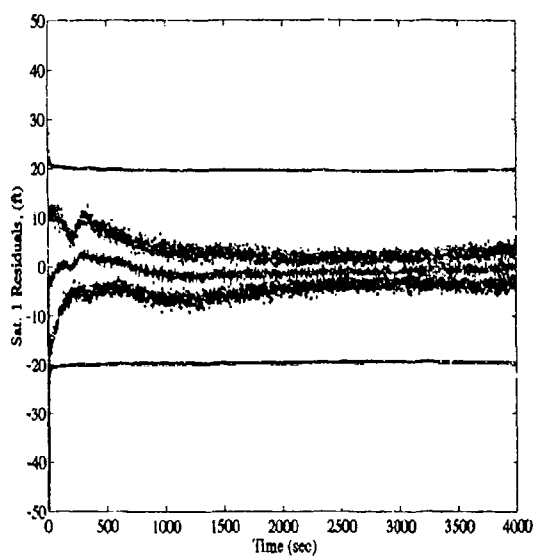


(Trans 4)

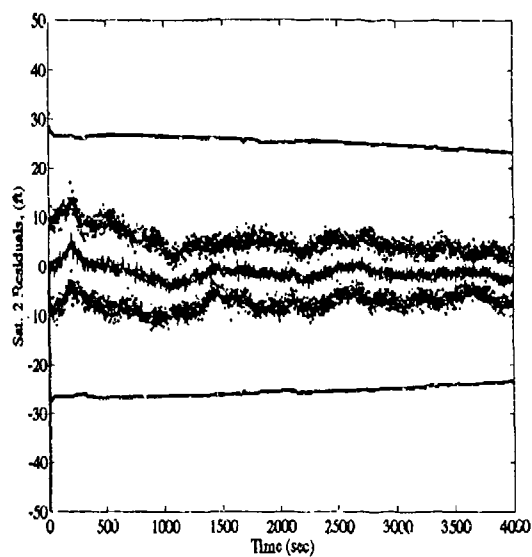


(Trans 5)

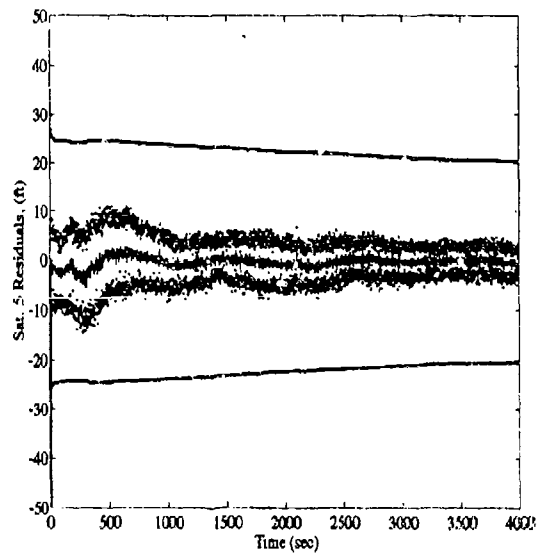
Figure H.4 NRS2 Transponder Scalar Residual Plots, Satellite Step Bias



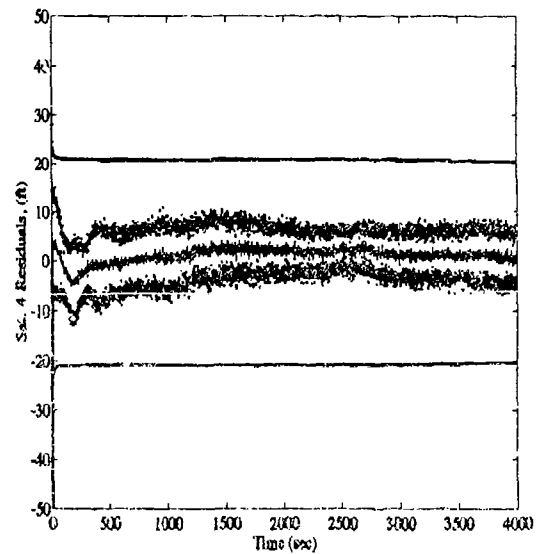
(Sat 1)



(Sat 2)

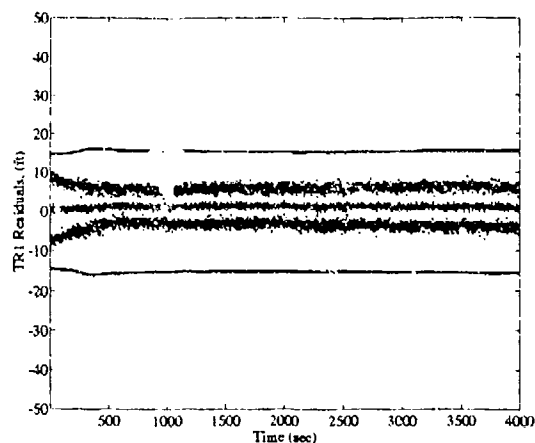


(Sat 5)

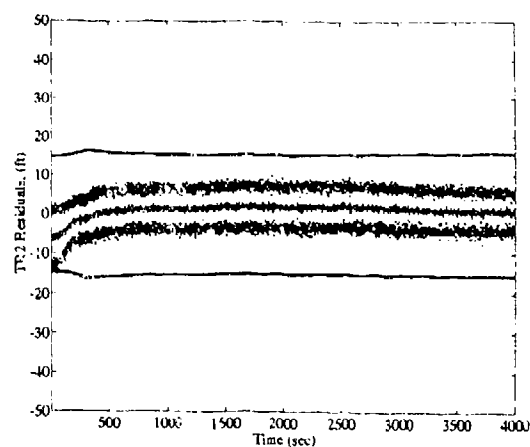


(Sat 4)

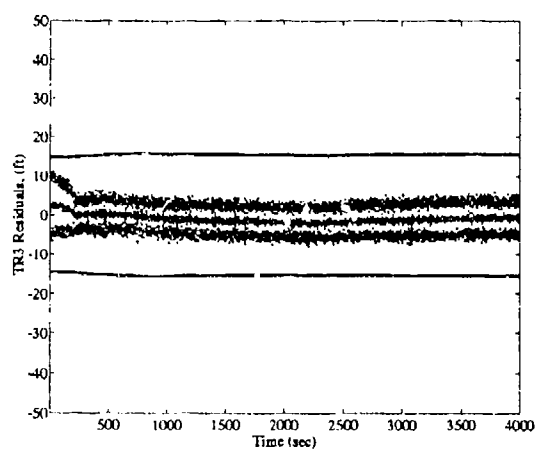
Figure H.5 NRS3 Satellite Scalar Residual Plots, Satellite Step Bias



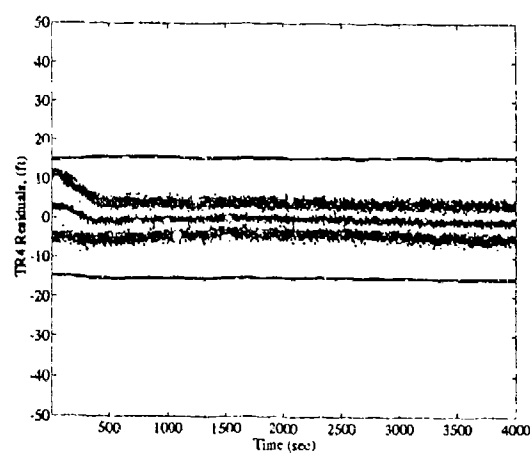
(Trans 1)



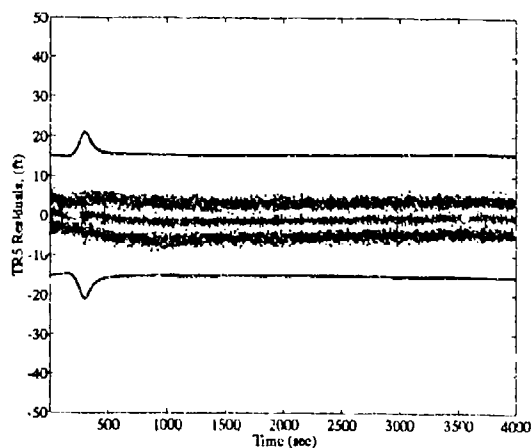
(Trans 2)



(Trans 3)

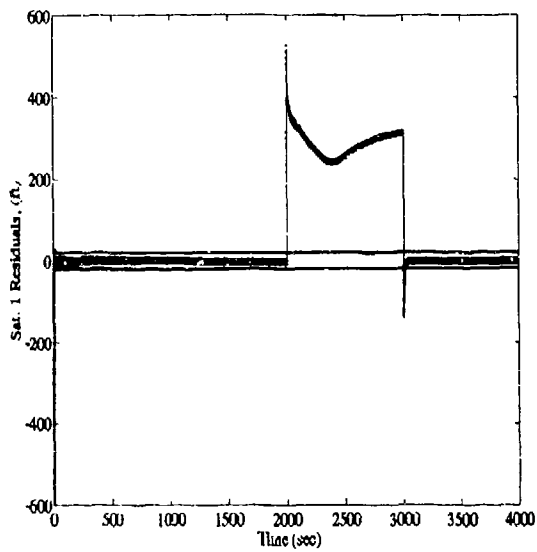


(Trans 4)

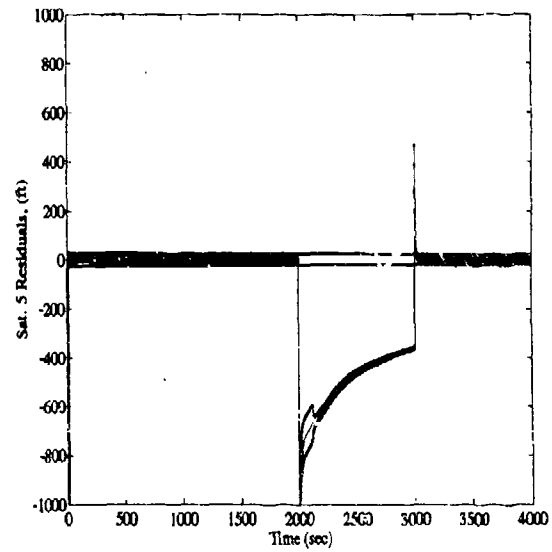


(Trans 5)

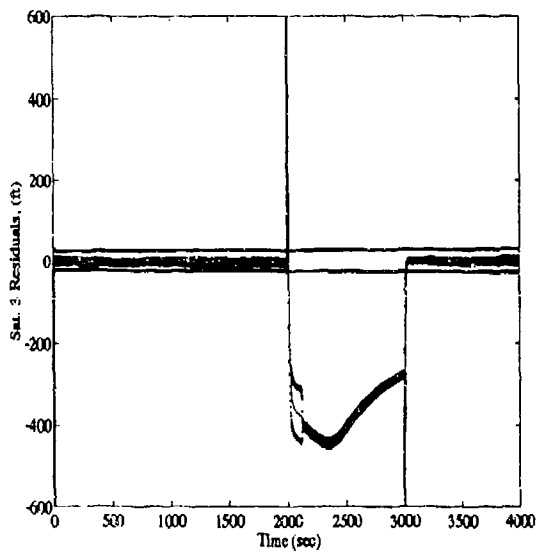
Figure H.6 NRS3 Transponder Scalar Residual Plots, Satellite Step Bias



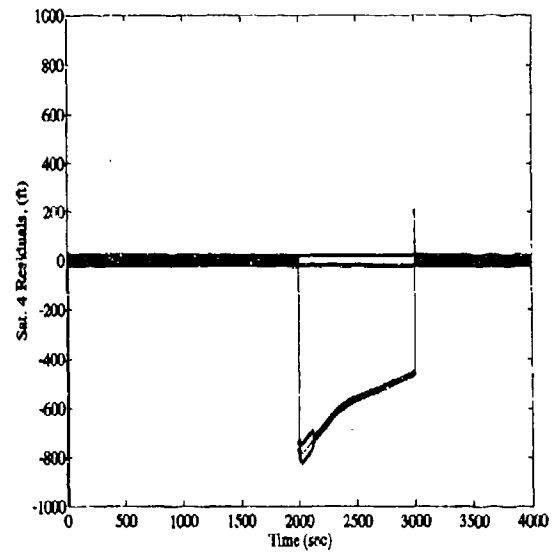
(Sat 1)



(Sat 5)

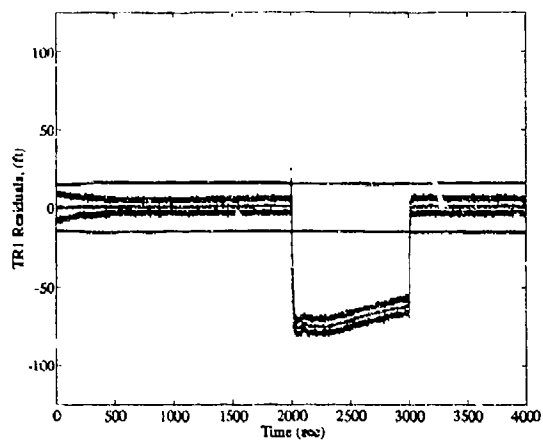


(Sat 3)

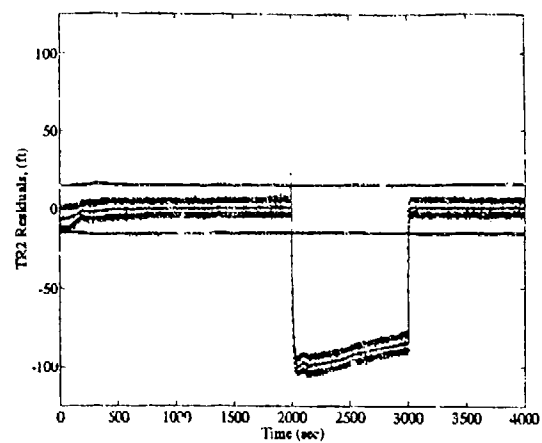


(Sat 4)

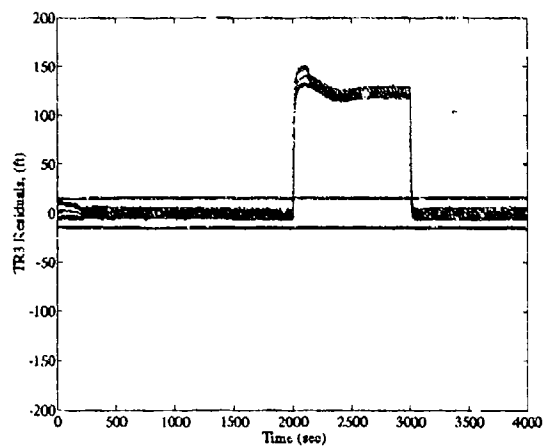
Figure H.7 NRS4 Satellite Scalar Residual Plots, Satellite Step Bias



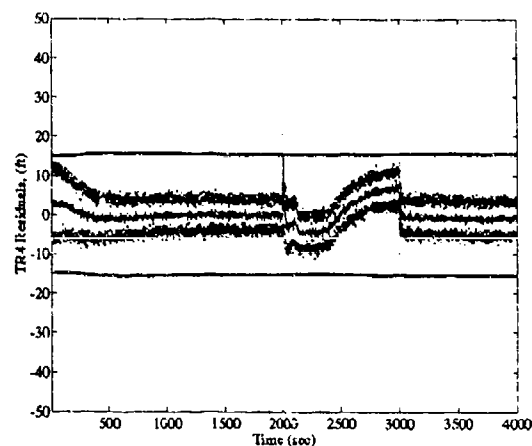
(Trans 1)



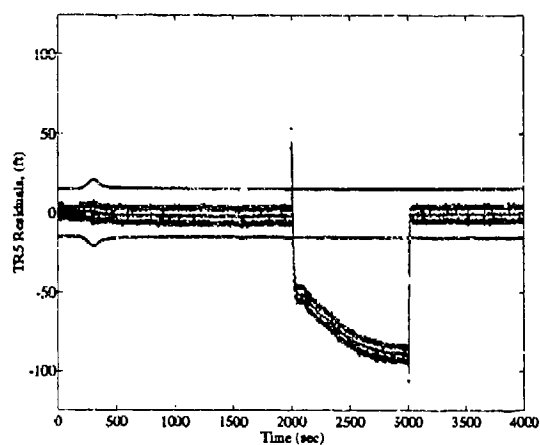
(Trans 2)



(Trans 3)

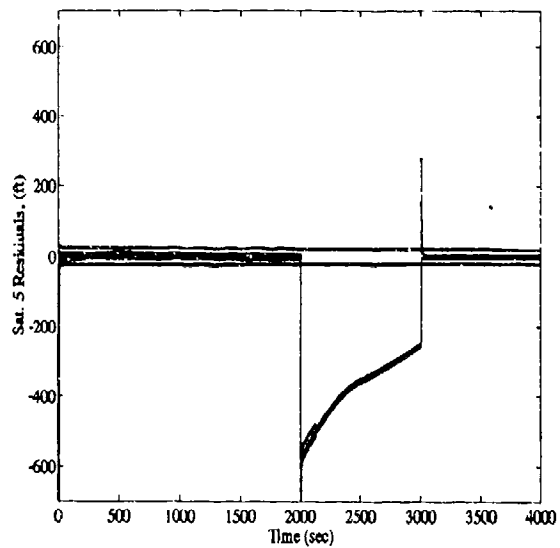


(Trans 4)

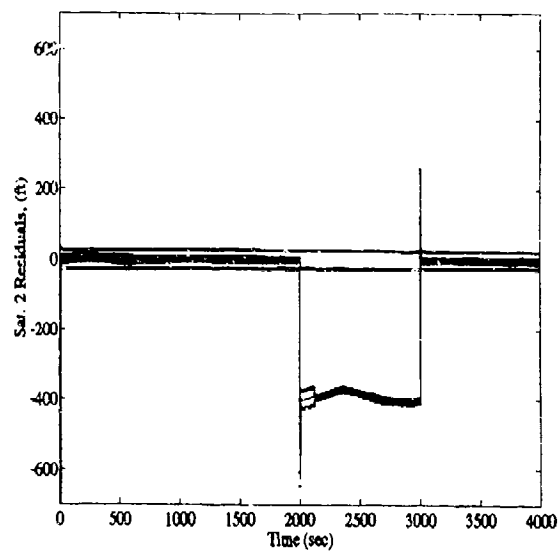


(Trans 5)

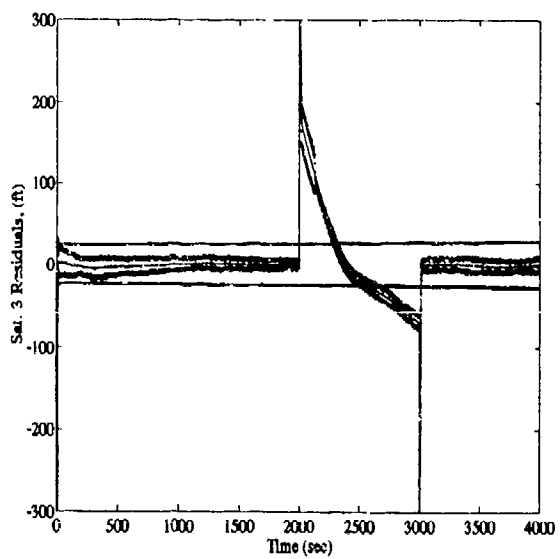
Figure H.8 NRS4 Transponder Scalar Residual Plots, Satellite Step Bias



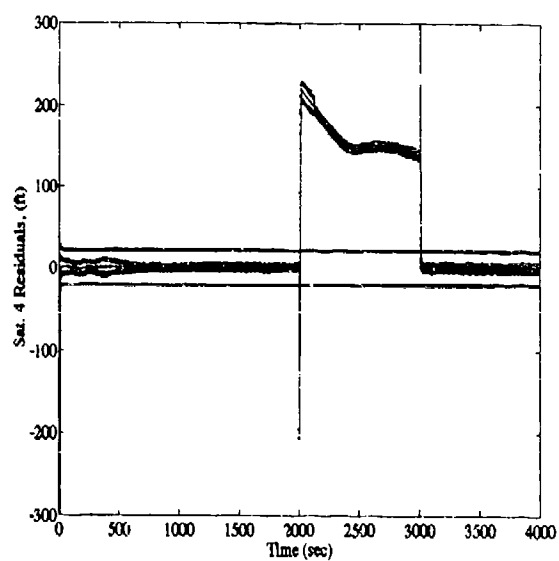
(Sat 5)



(Sat 2)

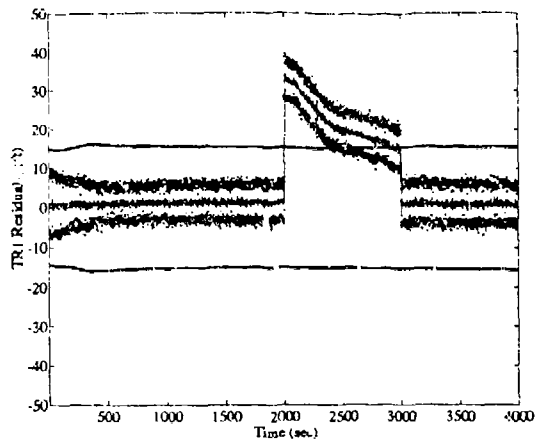


(Sat 3)

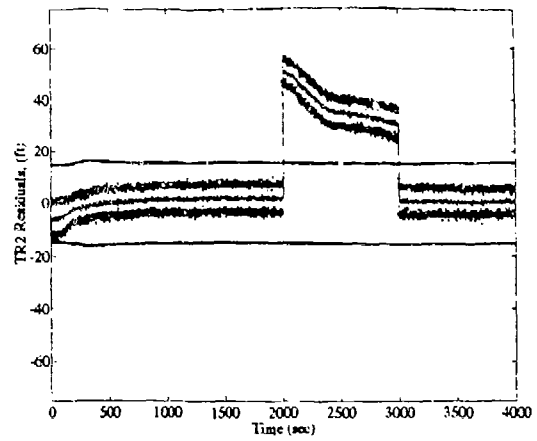


(Sat 4)

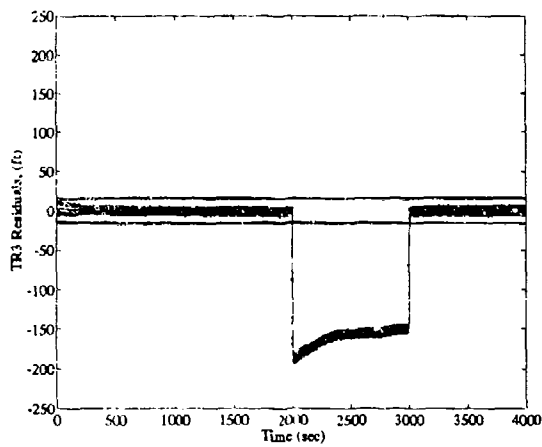
Figure H.9 NRS5 Satellite Scalar Residual Plots, Satellite Step Bias



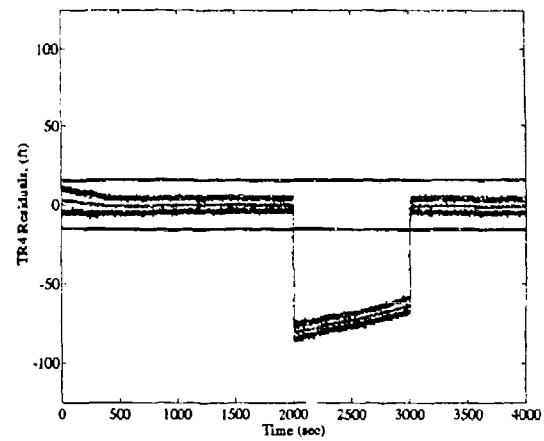
(Trans 1)



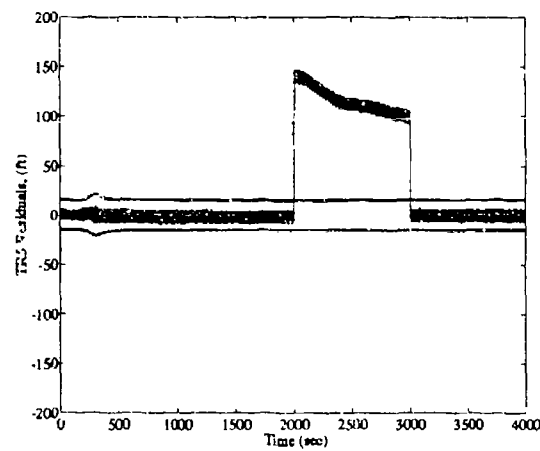
(Trans 2)



(Trans 3)

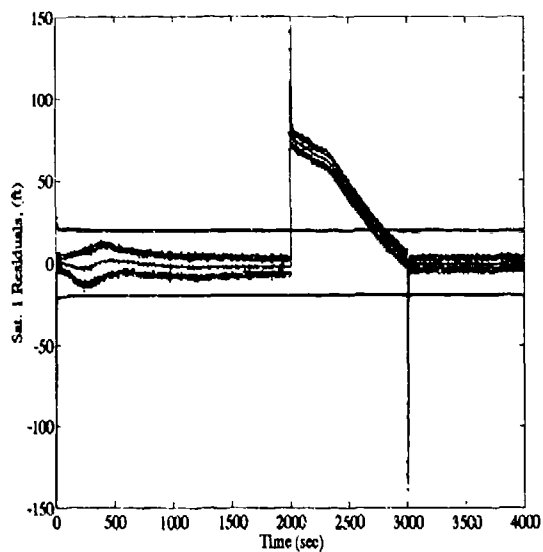


(Trans 4)

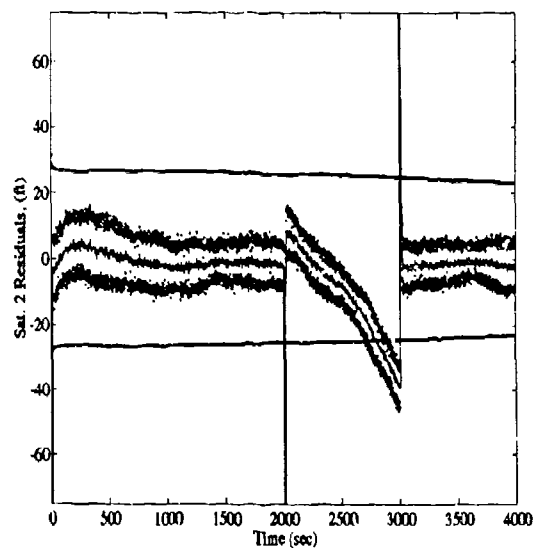


(Trans 5)

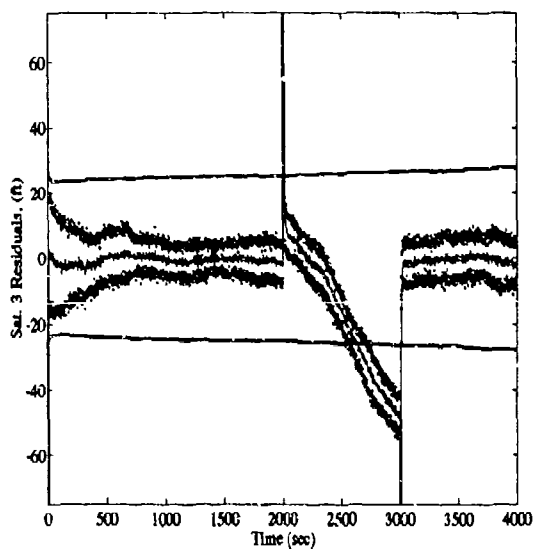
Figure H.10 NRS5 Transponder Scalar Residual Plots, Satellite Step Bias



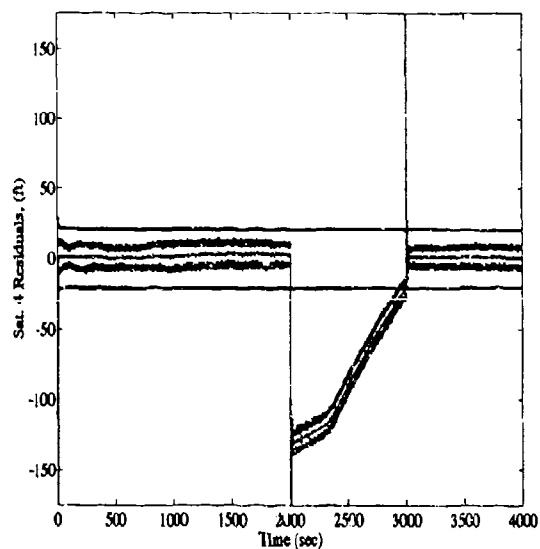
(Sat 1)



(Sat 2)

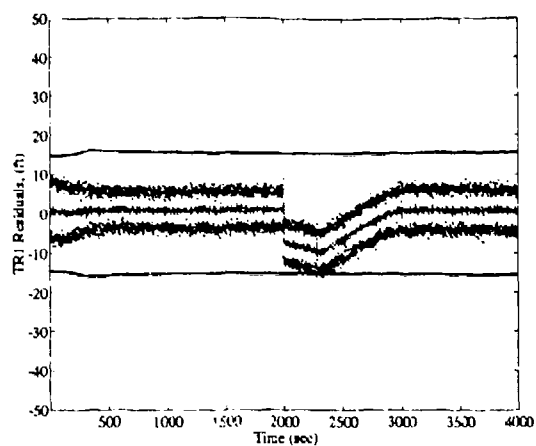


(Sat 3)

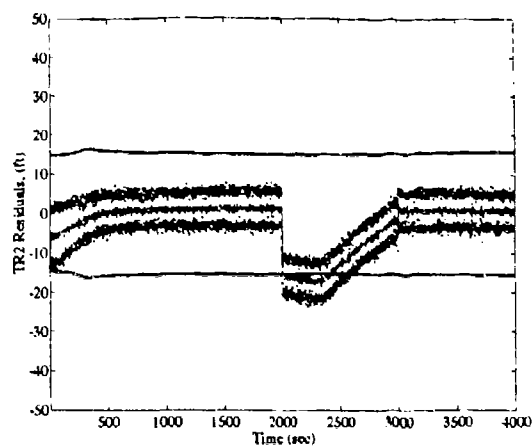


(Sat 4)

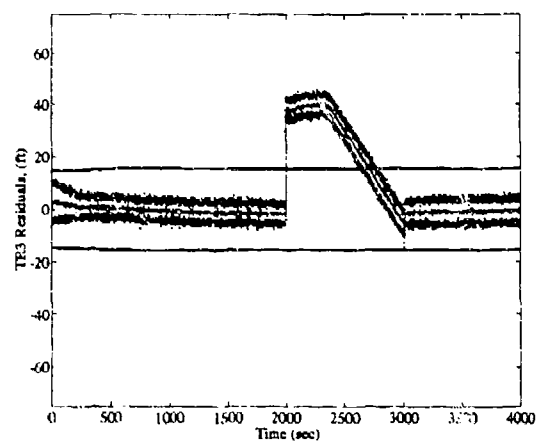
Figure H.11 NRS6 Satellite Scalar Residual Plots, Satellite Step Bias



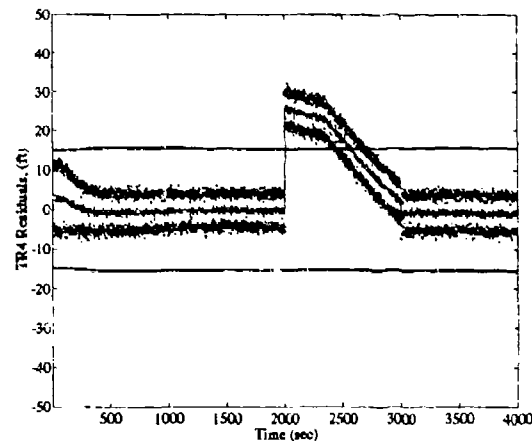
(Trans 1)



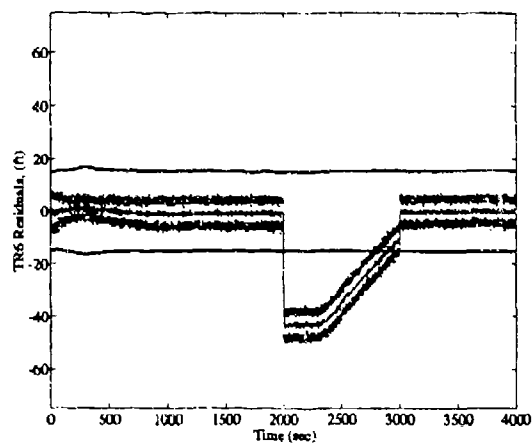
(Trans 2)



(Trans 3)

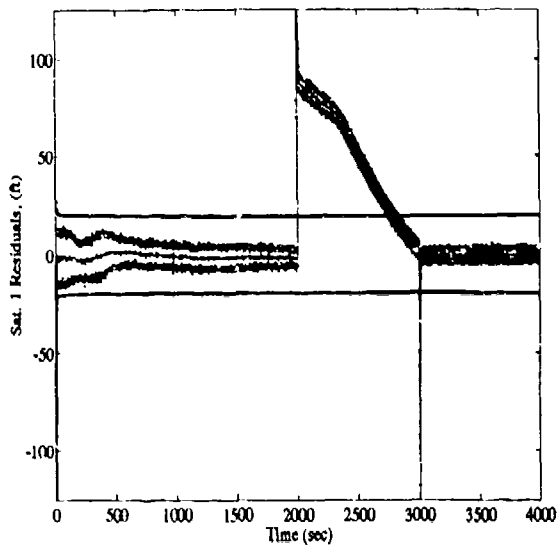


(Trans 4)

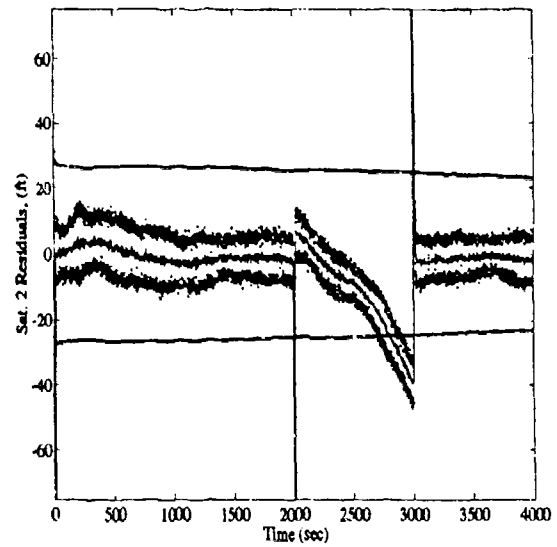


(Trans 6)

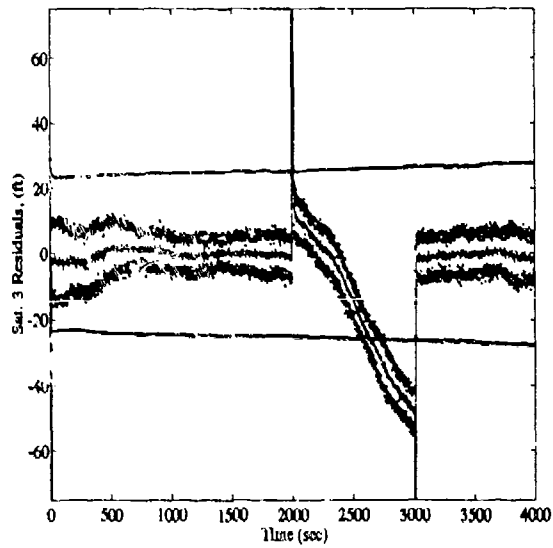
Figure H.12 NRS6 Transponder Scalar Residual Plots, Satellite Step Bias



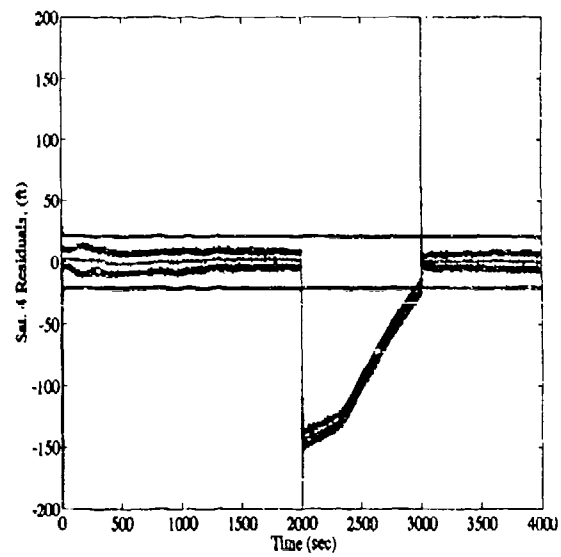
(Sat 1)



(Sat 2)

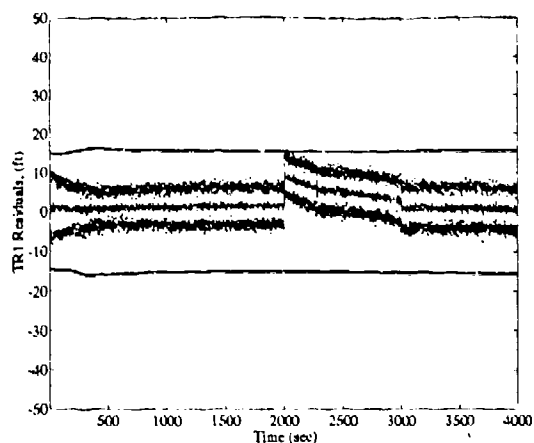


(Sat 3)

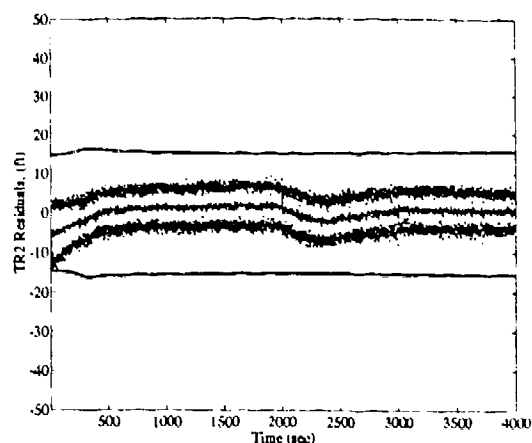


(Sat 4)

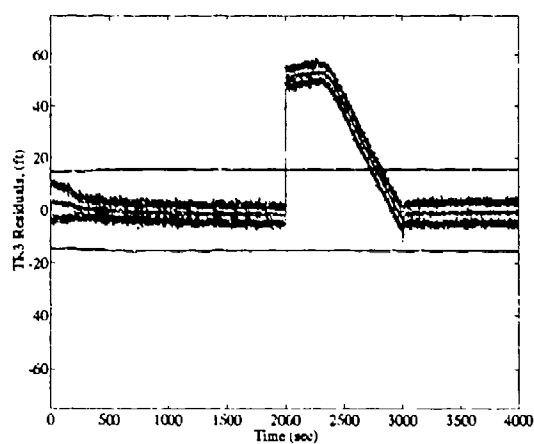
Figure H.13 NRS7 Satellite Scalar Residual Plots, Satellite Step Bias



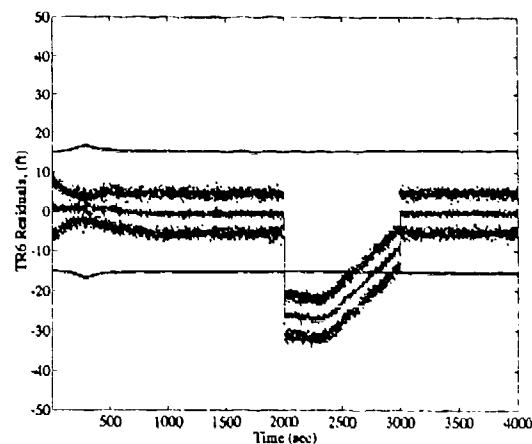
(Trans 1)



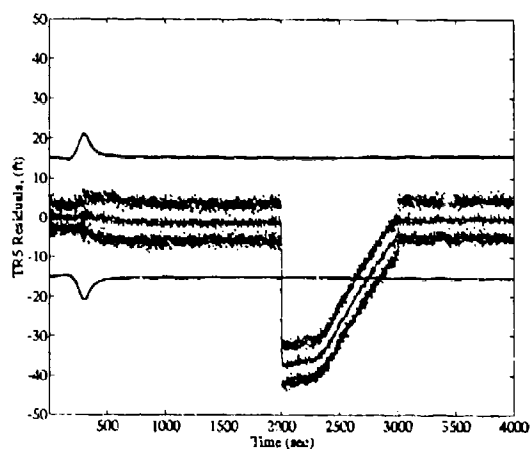
(Trans 2)



(Trans 3)

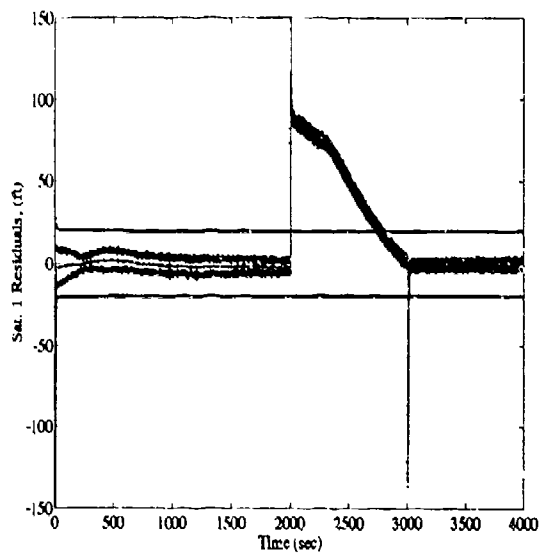


(Trans 6)

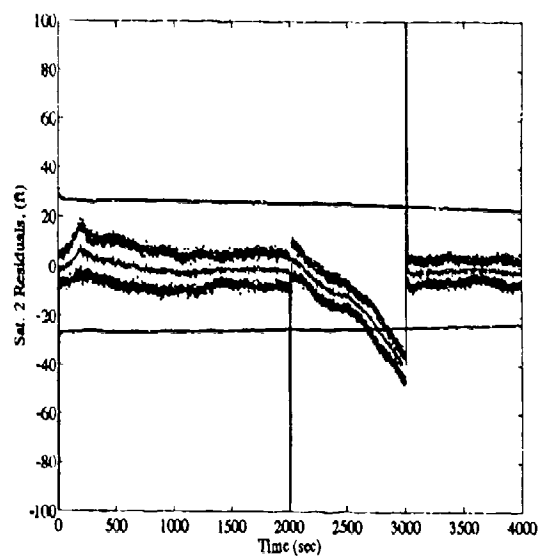


(Trans 7)

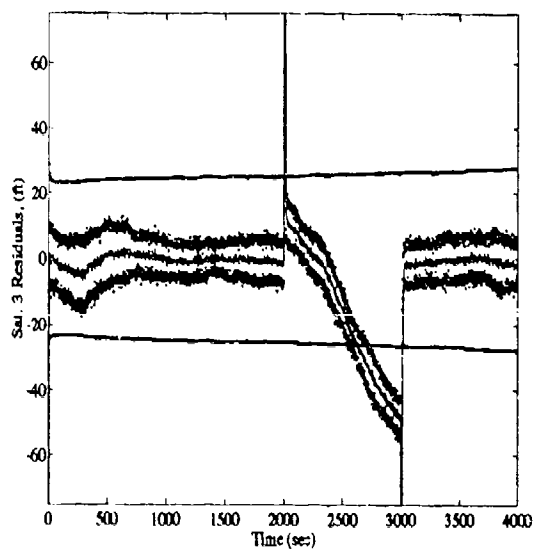
Figure H.14 NRS7 Transponder Scalar Residual Plots, Satellite Step Bias



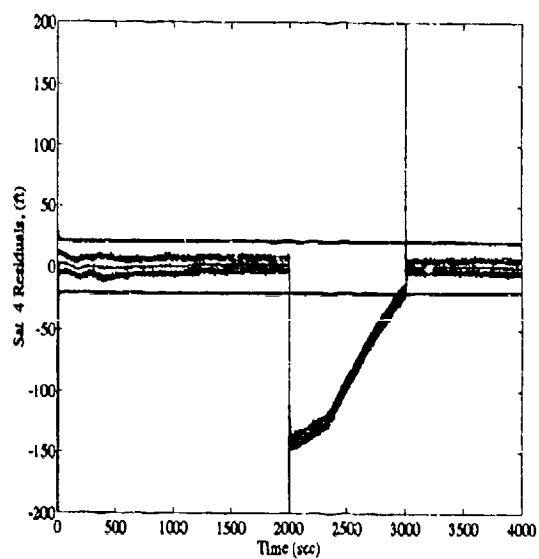
(Sat 1)



(Sat 2)

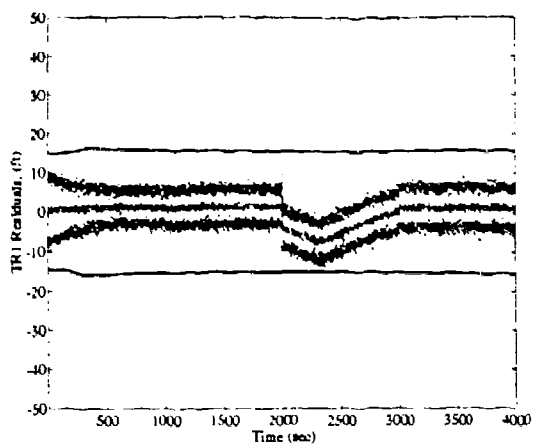


(Sat 3)

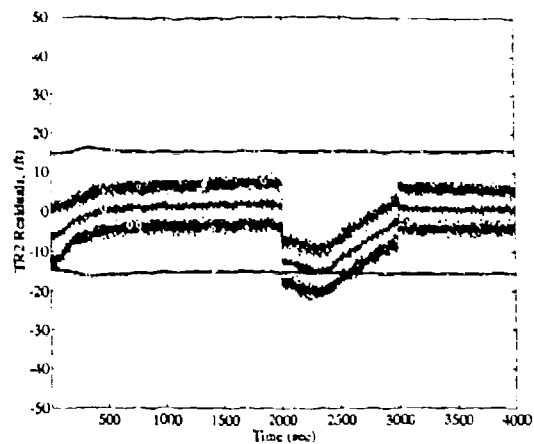


(Sat 4)

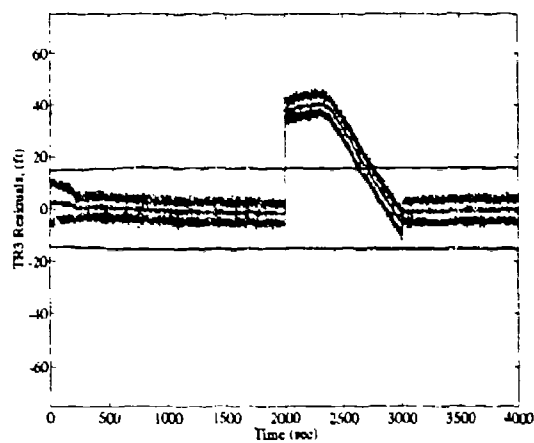
Figure H.15 NRS8 Satellite Scalar Residual Plots, Satellite Step Bias



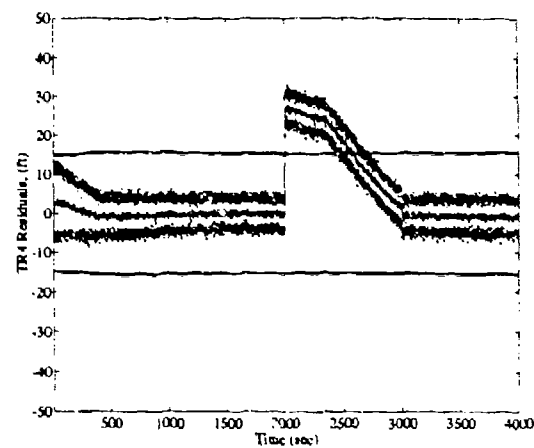
(Trans 1)



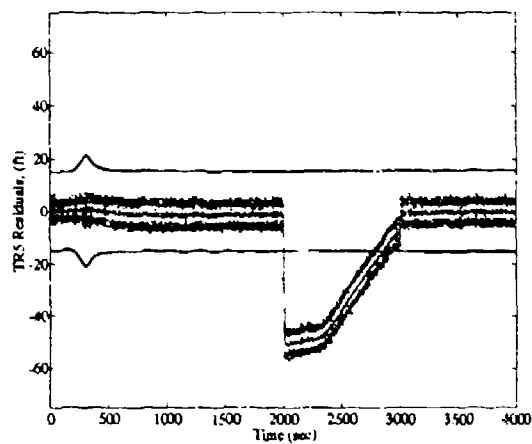
(Trans 2)



(Trans 6)

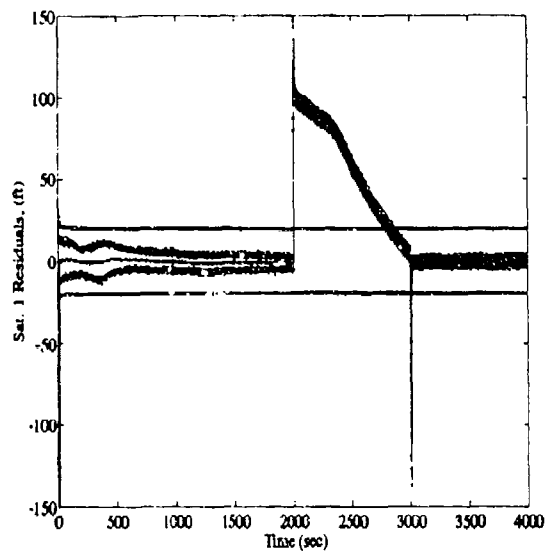


(Trans 4)

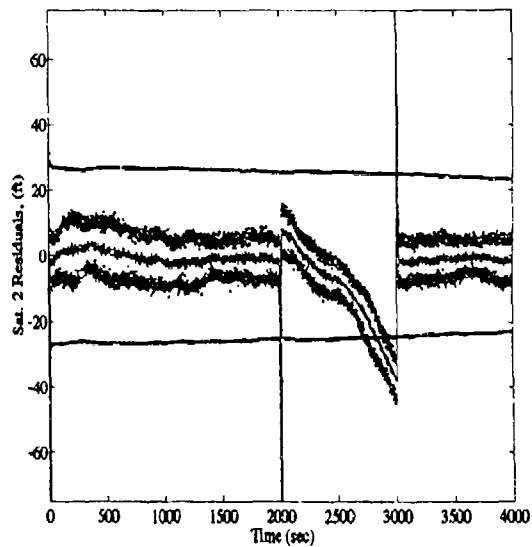


(Trans 5)

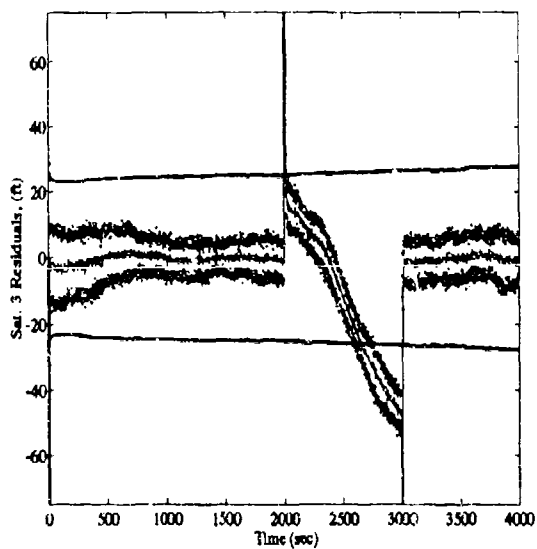
Figure H.16 NRS8 Transponder Scalar Residual Plots, Satellite Step Bias



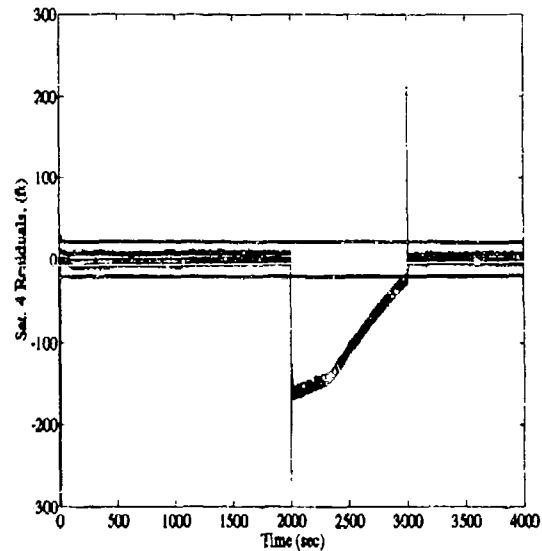
(Sat 1)



(Sat 2)

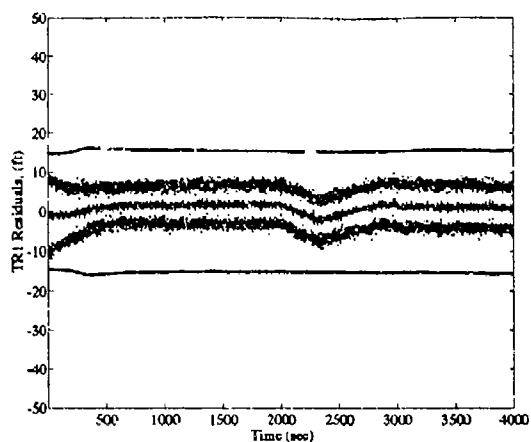


(Sat 3)

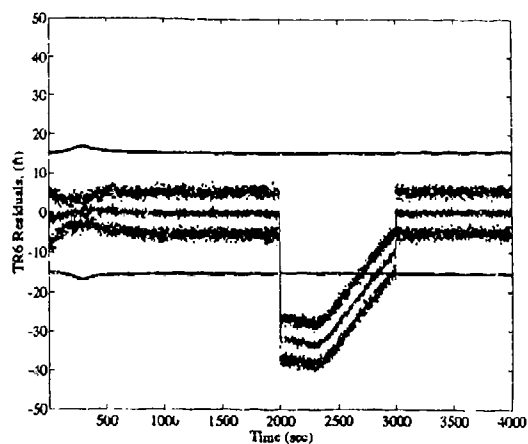


(Sat 4)

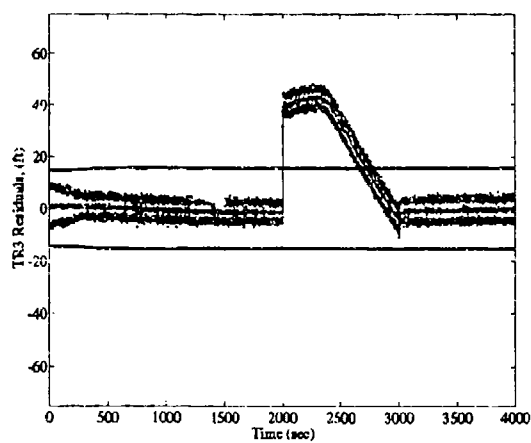
Figure H.17 NRS9 Satellite Scalar Residual Plots, Satellite Step Bias



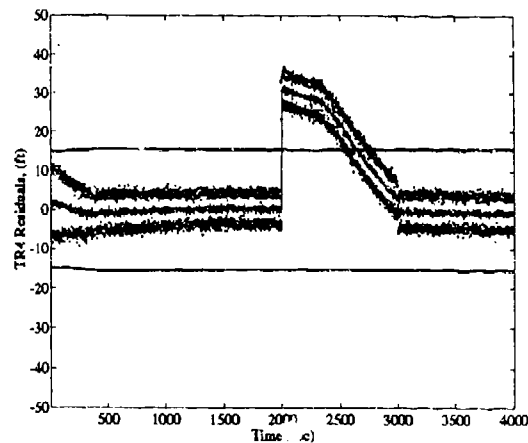
(Trans 1)



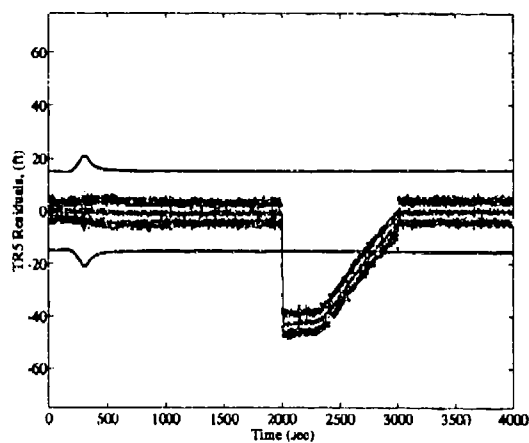
(Trans 6)



(Trans 3)

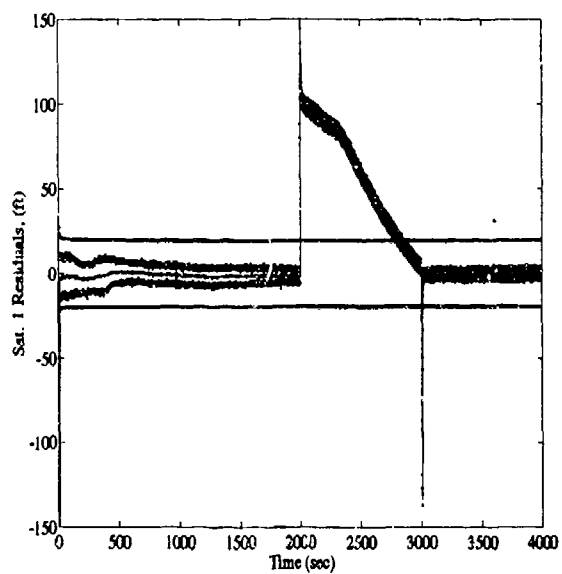


(Trans 4)

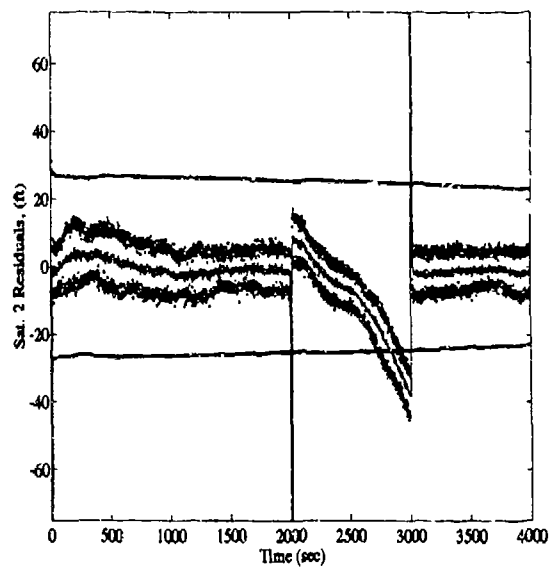


(Trans 5)

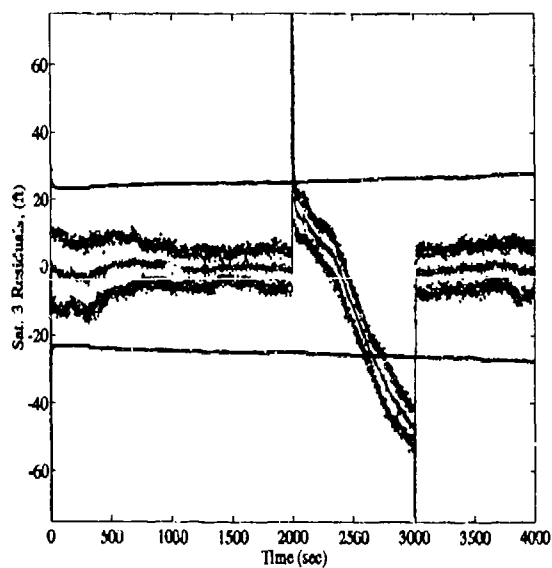
Figure H.18 NRS9 Transponder Scalar Residual Plots, Satellite Step Bias



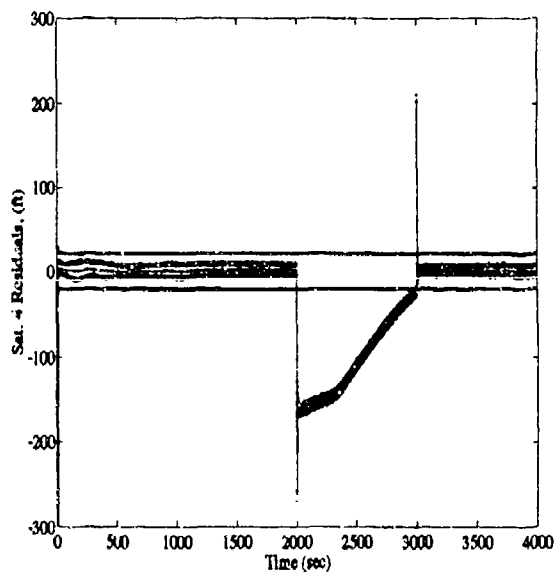
(Sat 1)



(Sat 2)

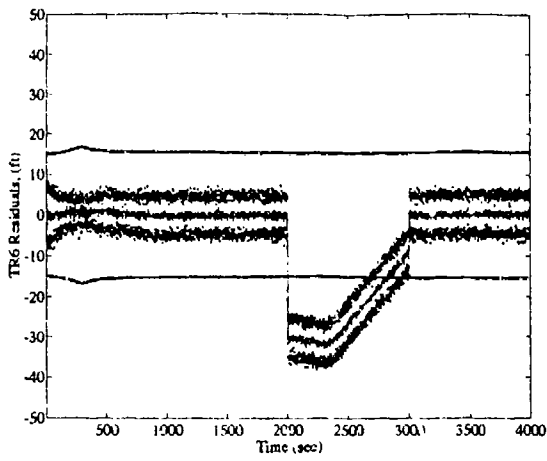


(Sat 3)

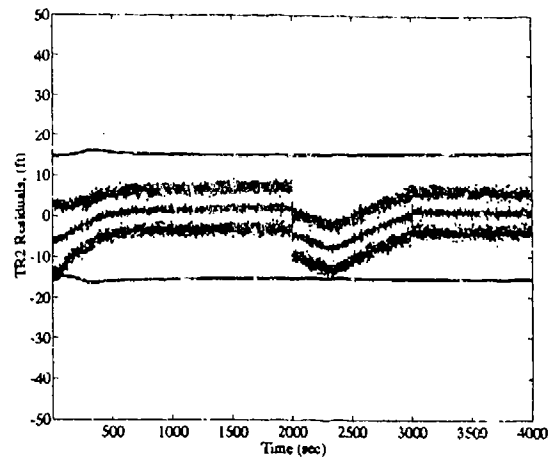


(Sat 4)

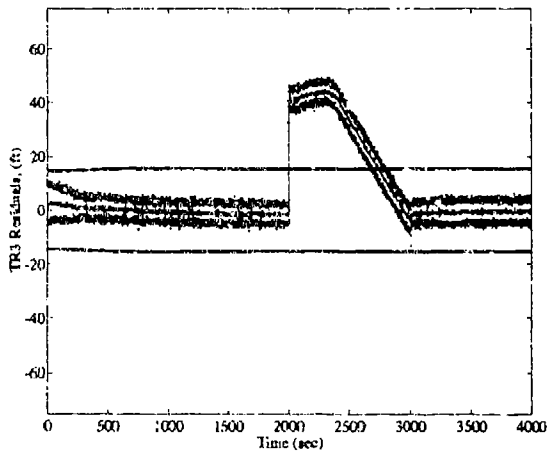
Figure H.19 NRS10 Satellite Scalar Residual Plots, Satellite Step Bias



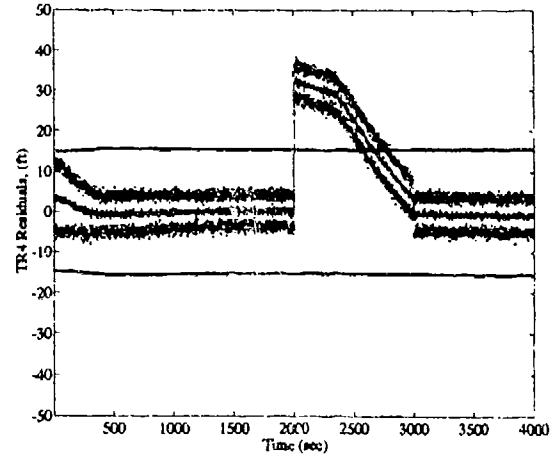
(Trans 6)



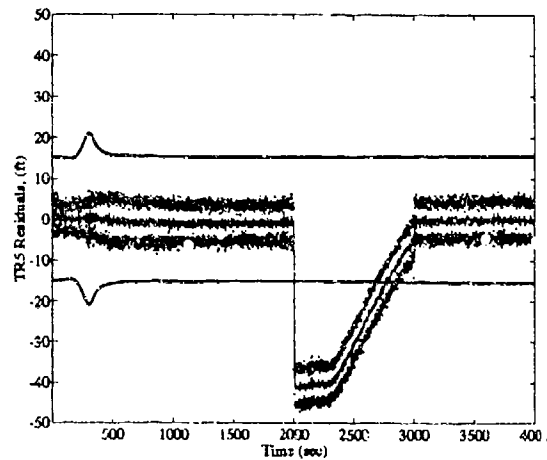
(Trans 2)



(Trans 3)



(Trans 4)



(Trans 5)

Figure H.20 NRS10 Transponder Scalar Residual Plots, Satellite Step Bias

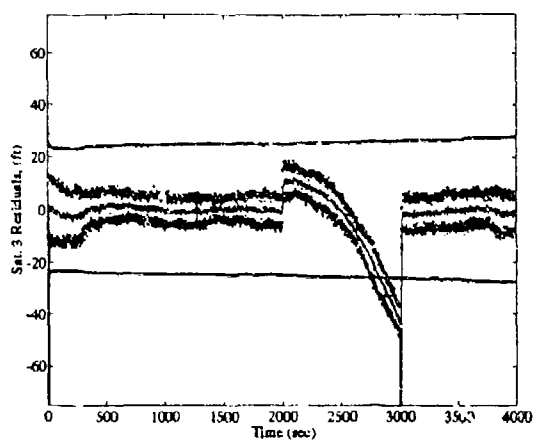
Appendix I. Residual Plots for a Satellite Signal Ramp Failure

This appendix contains limited scalar residual data for the ten NRS filters during a simulated ramp offset of slope $[4(t - 2000)]\text{ft}$ on the Satellite 3 pseudorange signal from time $t = 2000\text{sec}$ through $t = 3000\text{sec}$. The plots contained in this section are identical in format to those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

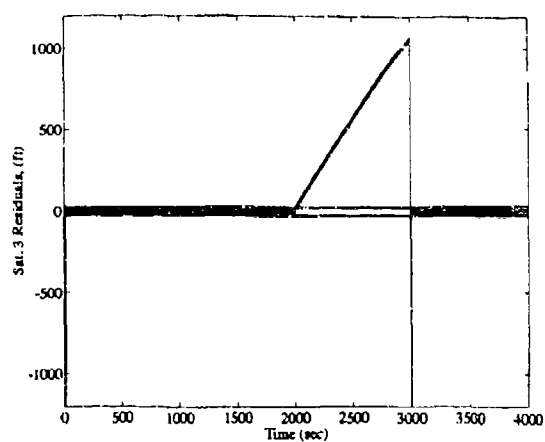
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. Unlike Appendices D, E, and H, this section contains only the scalar residual plots for the failed measurement signal (Satellite 3) in nine of the ten filters. Filter 3 is not included, because it does not receive the failed measurement. The first plot presented is for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS10. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table I.1 Legend for Filter Tuning Plots

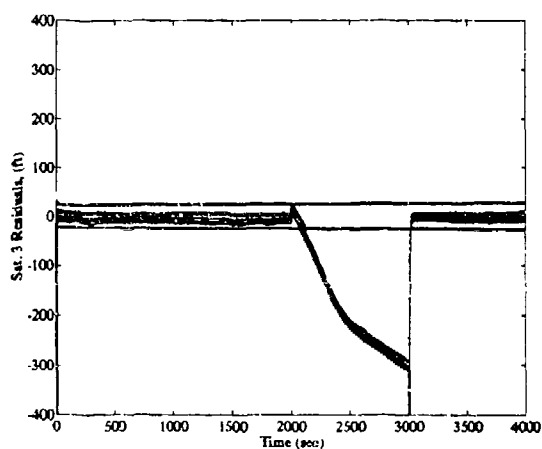
Symbol	Definition
- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
-- Dashed Line	\pm Filter-Predicted Sigma



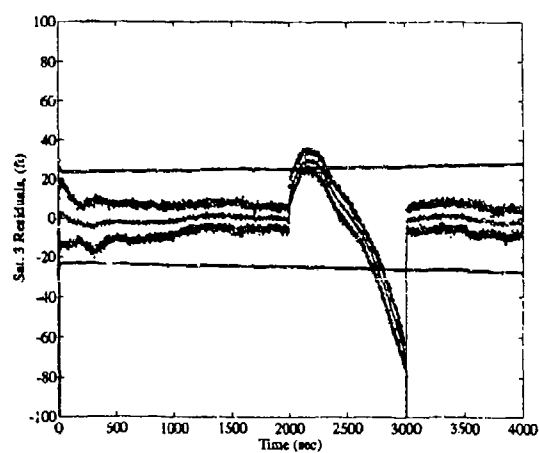
(Filter 1)



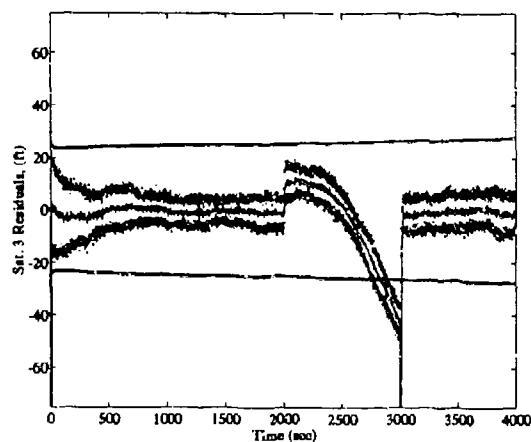
(Filter 2)



(Filter 4)

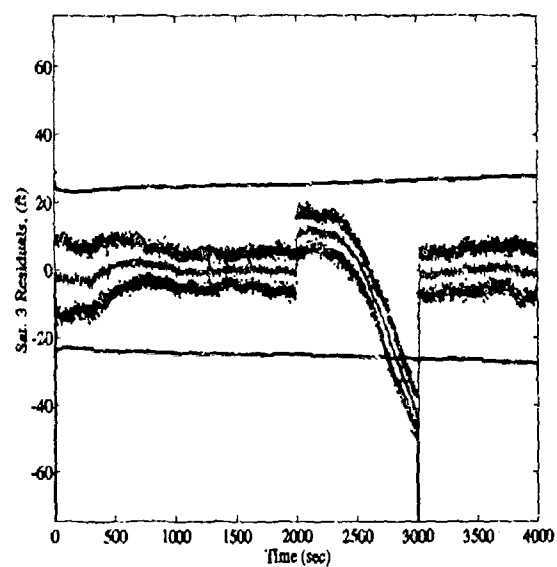


(Filter 5)

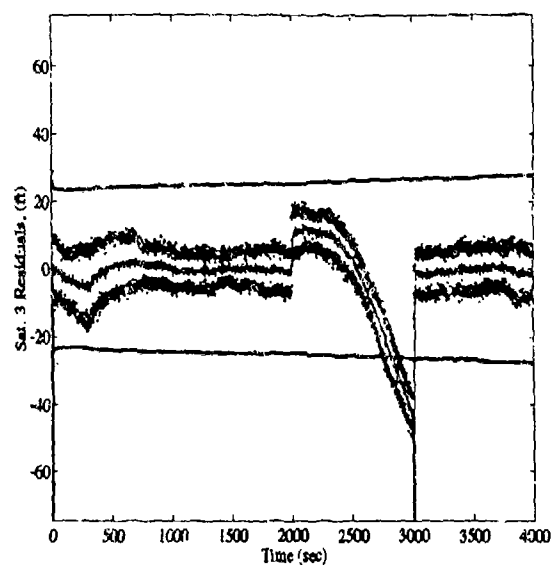


(Filter 6)

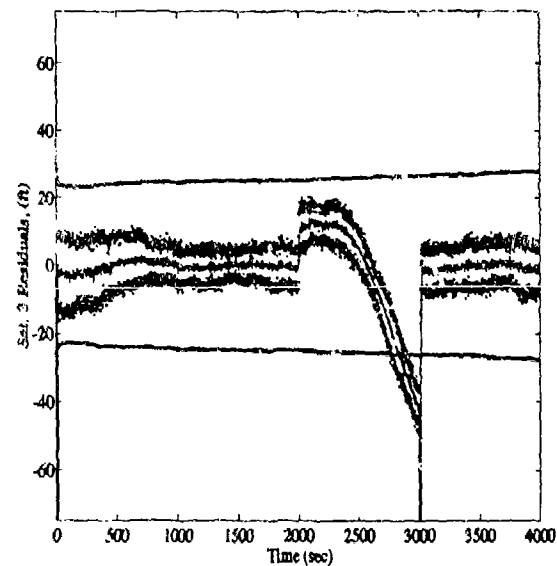
Figure I.1 Satellite 3 Scalar Residual Plots, Satellite Ramp Bias for NRS Filters 1,2,4,5,6



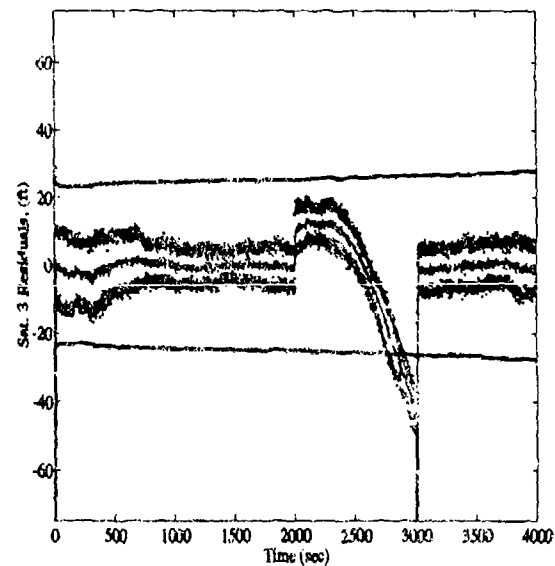
(Filter 7)



(Filter 8)



(Filter 9)



(Filter 10)

Figure I.2 Satellite 3 Scalar Residual Plots, Satellite Ramp Bias for NRS Filters 7-10

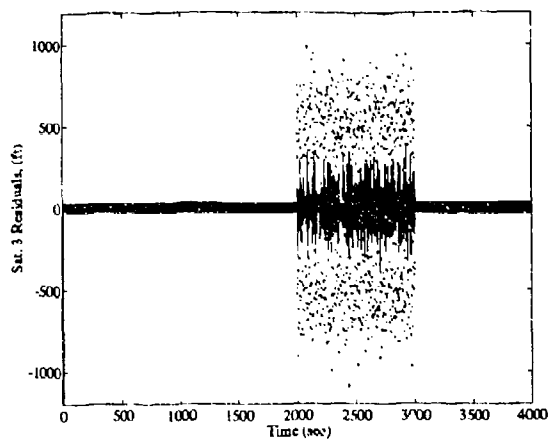
Appendix J. Residual Plots for a Satellite Signal Noise Failure

This appendix contains limited scalar residual data for the ten NRS filters during a simulated noise increase of (300v)ft on the Satellite 3 pseudorange signal from time $t = 2000sec$ through $t = 3000sec$. The plots contained in this section are identical in format to those in Appendix D. These plots are presented to support the validity of the MNRS as a valid FDIR algorithm for an integrated navigation system.

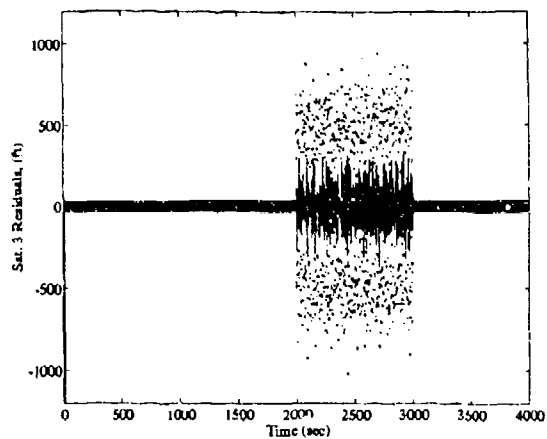
A legend is provided for quick reference as to which of the lines on the graph corresponds to which of the variables of interest. Unlike Appendices D, E, and H, this section contains only the scalar residual plots for the failed measurement signal (Satellite 3) in nine of the ten filters. Filter 3 is not included, because it does not receive the failed measurement. The first plot presented is for the NRS1 filter. The plots follow in order for the other nine filters, the last being NRS10. Scalar residual plots are not presented for the velocity or the barometric altimeter measurement updates, since these are not used in the failure detection algorithm.

Table J.1 Legend for Filter Tuning Plots

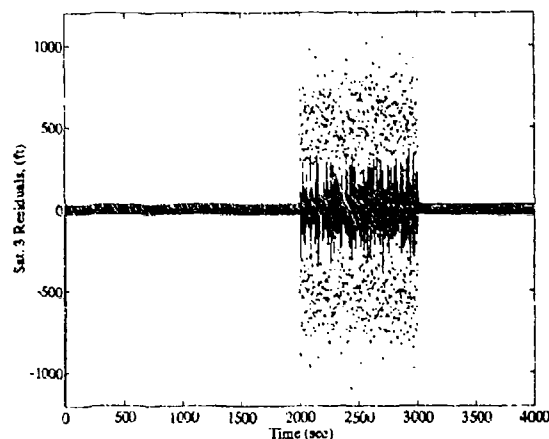
Symbol	Definition
-- Solid Line	Mean Error
... Dotted Line	Mean Error \pm True Sigma
- - Dashed Line	\pm Filter-Predicted Sigma



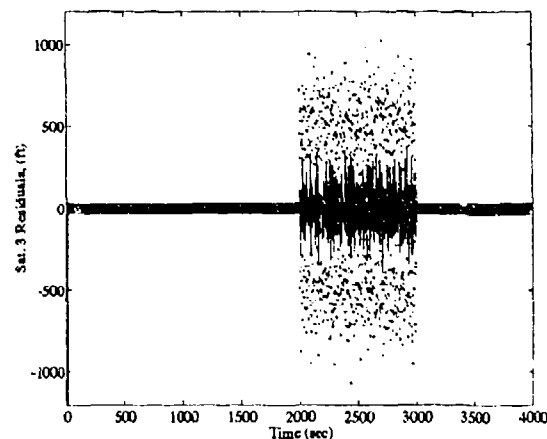
(Filter 1)



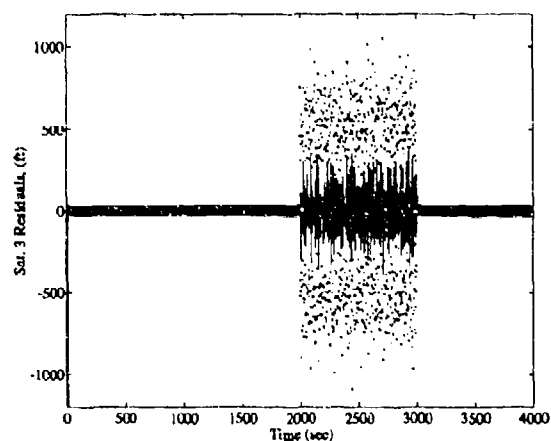
(Filter 2)



(Filter 4)

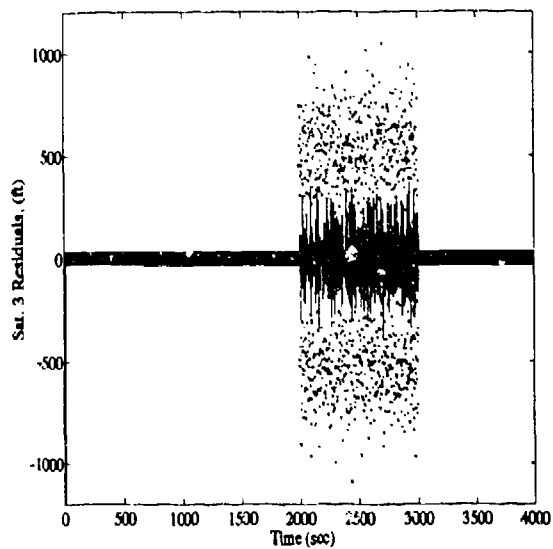


(Filter 5)

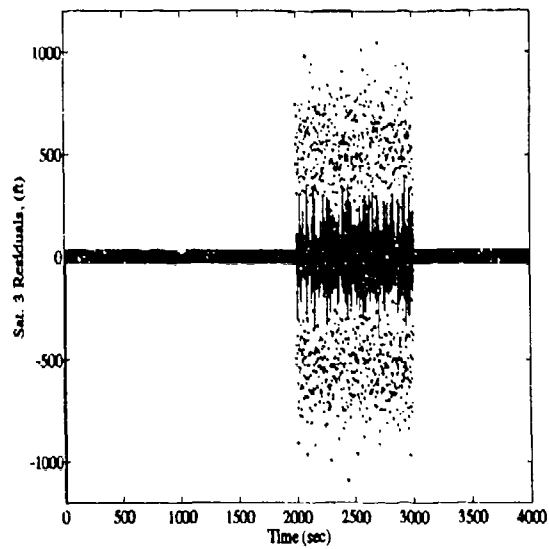


(Filter 6)

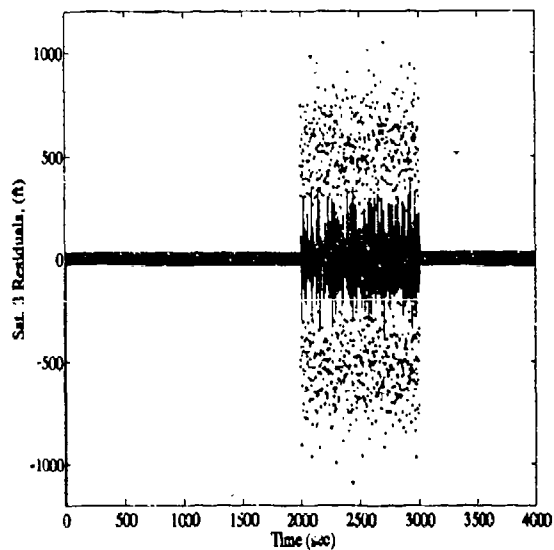
Figure J.1 Satellite 3 Scalar Residual Plots, Satellite Noise Increase for NRS Filters 1,2,4,5 and 6



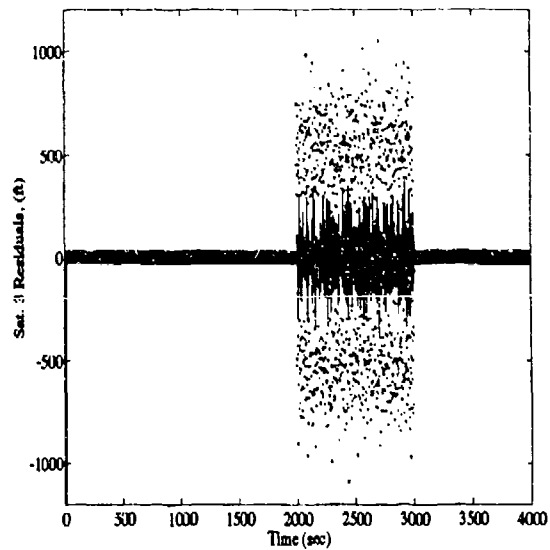
(Filter 7)



(Filter 8)



(Filter 9)



(Filter 10)

Figure J.2 Satellite 3 Scalar Residual Plots, Satellite Noise Increase for NRS Filters 7-10

Appendix K. Results of the NRS Filter Chi-Square Tests

Within the MNRS failure detection algorithm, the Chi-Square test signals when a failure occurs in any of the NRS filters. As described previously, a Chi-Square test is conducted on the residuals of each of the ten NRS filters. The occurrence of a measurement failure in a particular filter is determined by comparing the magnitude of the Chi-Square variable to an established failure threshold. If the Chi-Square variable value exceeds the threshold, a measurement failure is declared and the MNRS looks to a different NRS filter for its navigation solution.

The following plots portrays this algorithm completely for each failure. The plots are organized according to failure run, beginning with the baseline run and concluding with the noise failure on transponder 1. For each of the failure runs the following plots are presented.

First the individual Chi-Square tests on the ten NRS filters are displayed. These plots compare the Chi-Square magnitude,

$$\chi(t_k) = \sum_{j=k-2+1}^k \gamma^T(t_j) \Lambda^{-1}(t_j) \gamma(t_j) \quad (\text{K.1})$$

with a predetermined threshold, $T = 15$. The window size of the Chi-Square test was chosen to be $N = 3$. The Chi-Square threshold and the window size have been chosen for good performance (minimize false alarm rate and detection delay, while maximizing failure detection rate). It should also be noted that the residual and covariance matrices, $\gamma(t_j)$ and $\Lambda(t_j)$ respectively, contain only the measurement information corresponding to the four satellite and five transponder updates. Following the Chi-Square plots, the final two plots show when the individual Chi-Square tests fail the threshold test. The first five filters for each run are on the first plot, followed by the last five filters on the second plot. When an individual filter's Chi-Square test exceeds the threshold, the line corresponding to that filter will switch high. The filter is considered to be navigating accurately, according to the Chi-Square test, when the line is low.

These plots have been included to demonstrate the robust nature of the MNRS filter to single failures. These plots also demonstrate the ability for the MNRS to determine

which of the ten filters is producing the accurate navigation solution during single measurement failures.

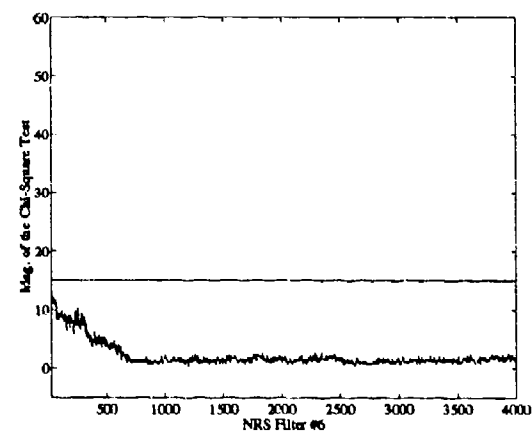
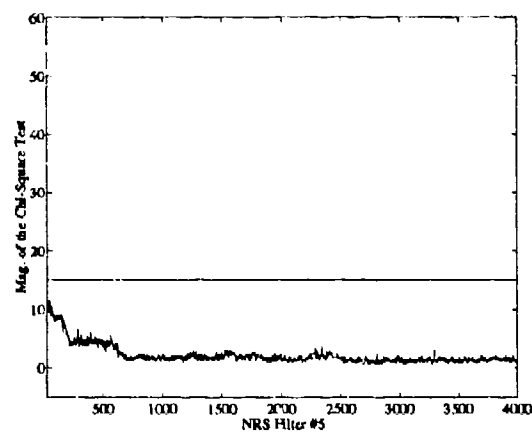
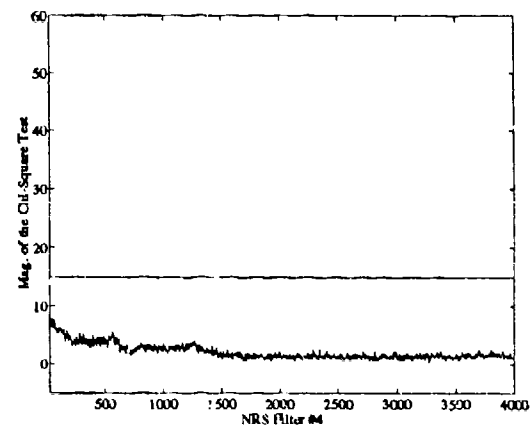
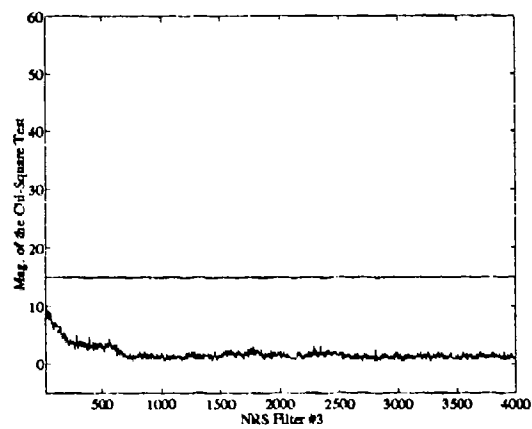
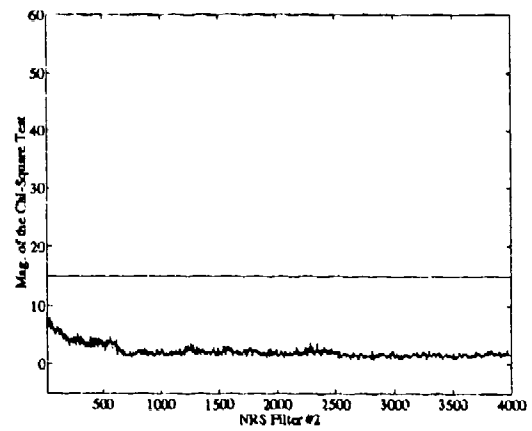
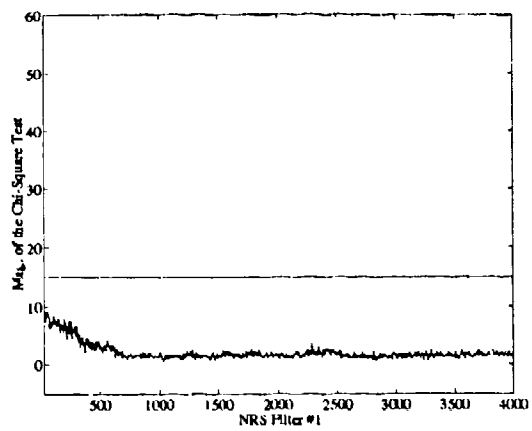


Figure K.1 Baseline Run, Chi-Square Test Results for Filters 1-6, N=3

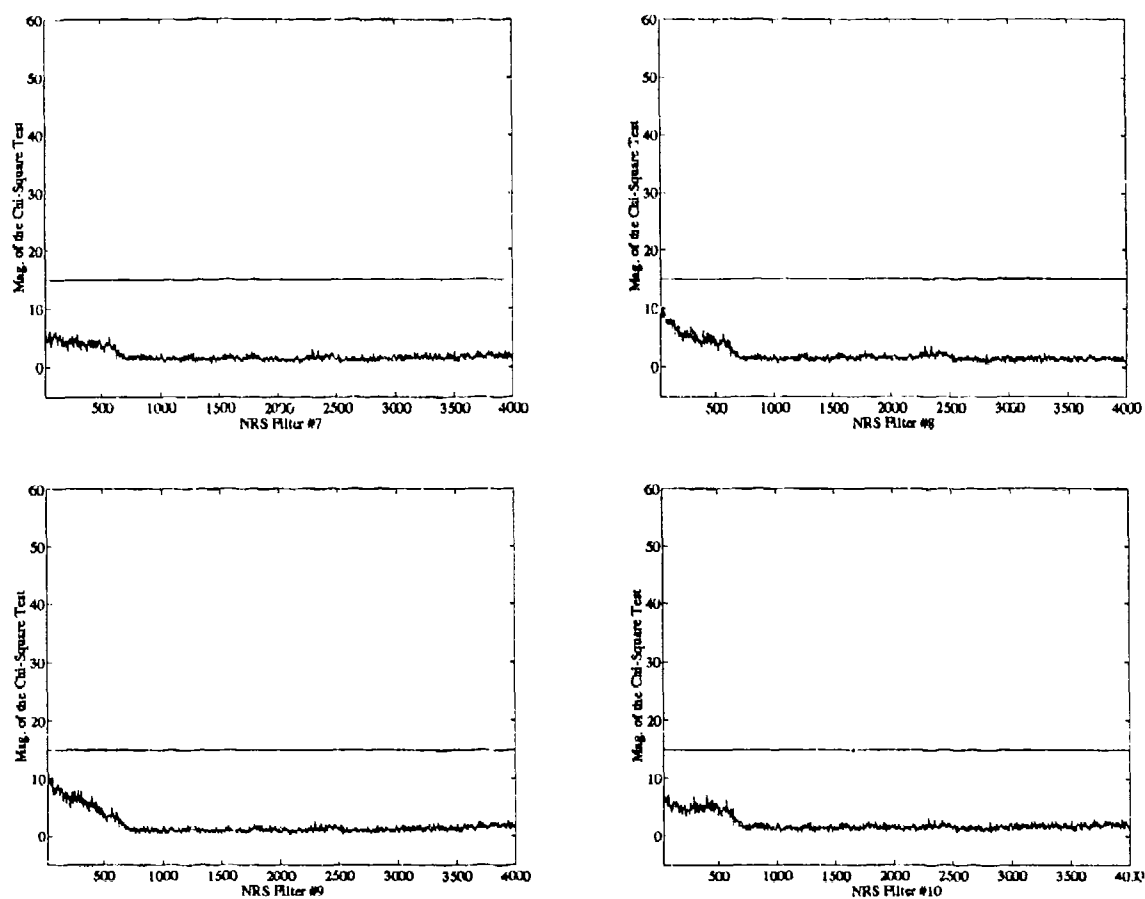


Figure K.2 Baseline Run, Chi-Square Test Results for Filters 7-10, N=3

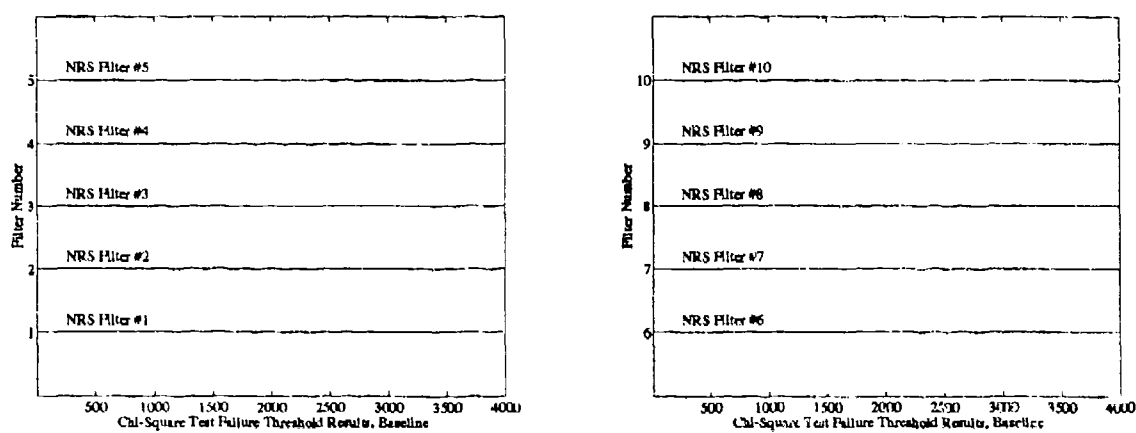


Figure K.3 Threshold Test Results for the Chi-Square Tests on each Filter, Baseline Run

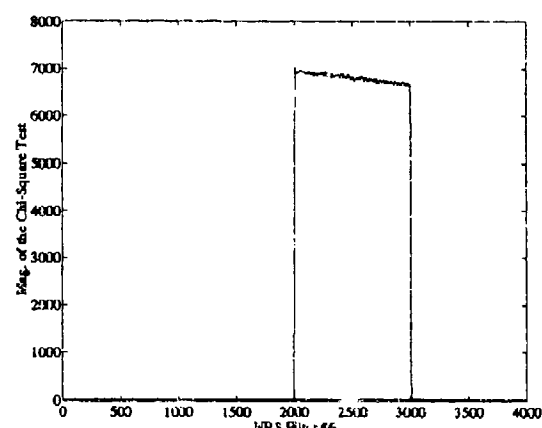
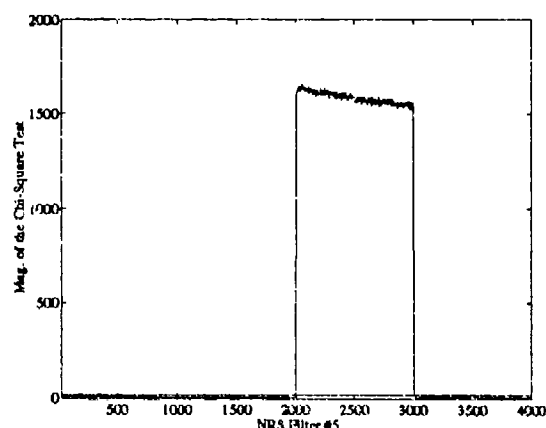
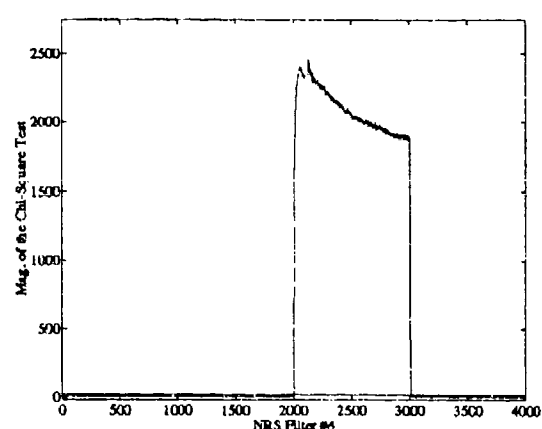
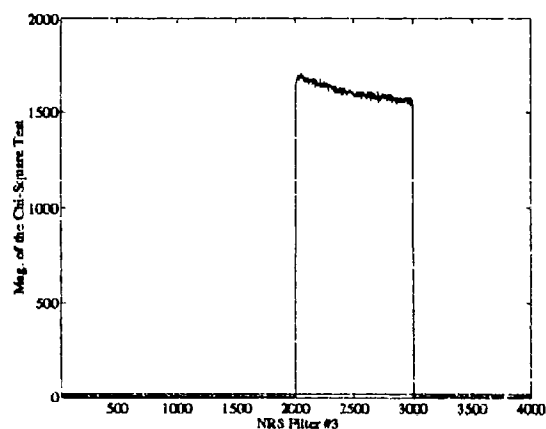
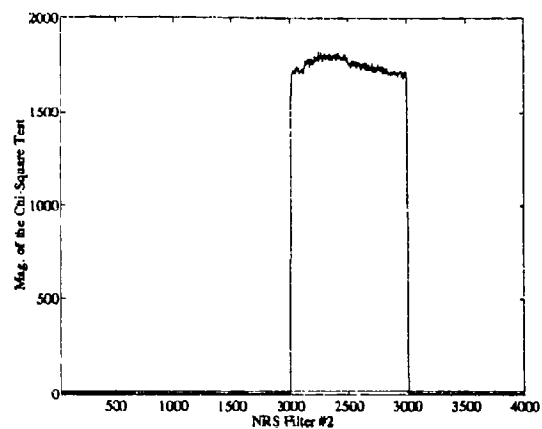
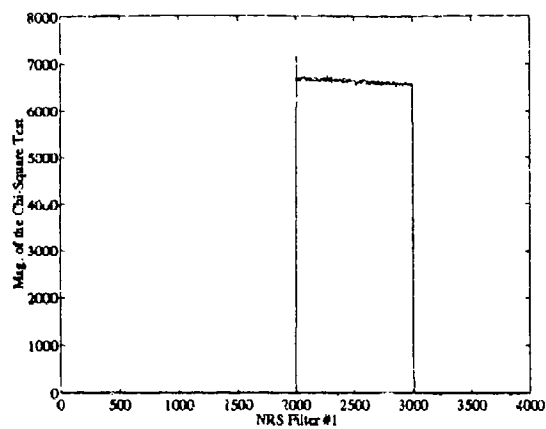


Figure K.4 Transponder 1 *Step Bias*=800, Chi-Square Test Results for Filters 1-6, N=3

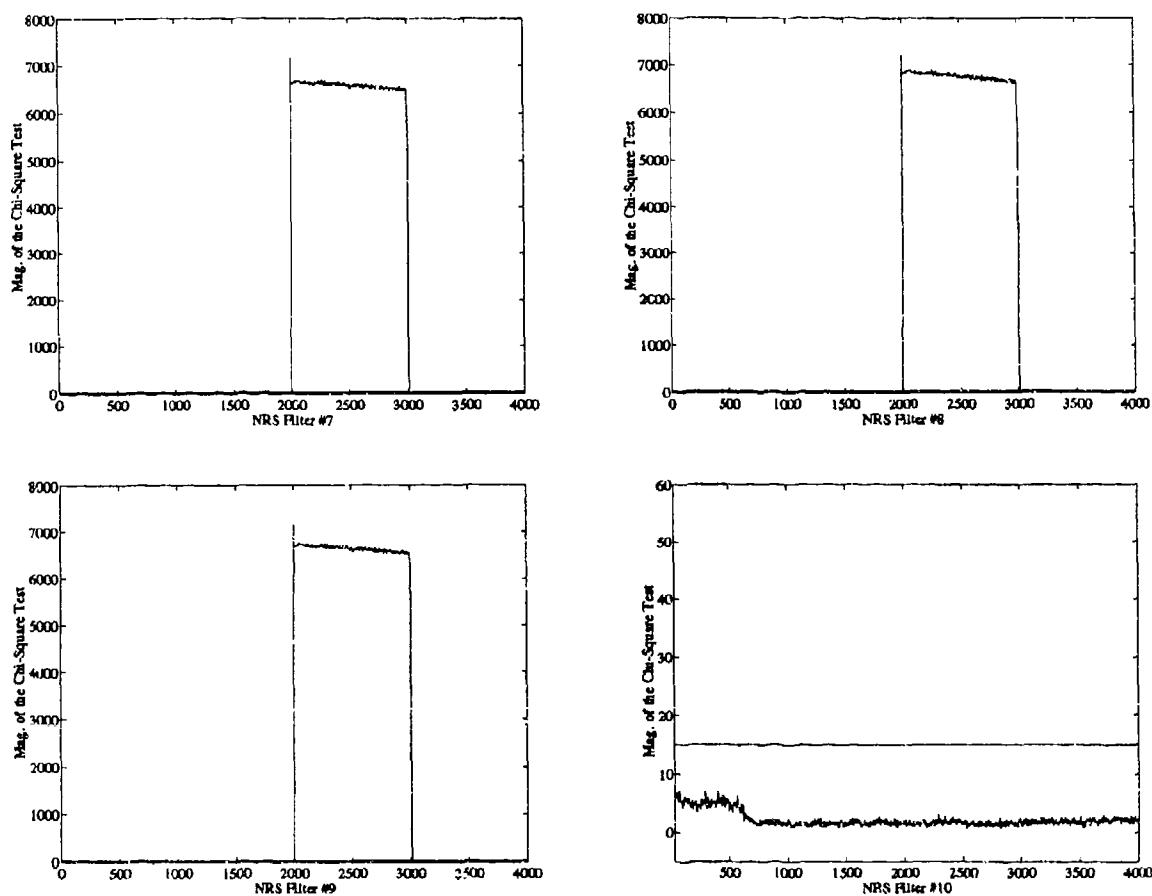


Figure K.5 Transponder 1 *Step Bias*=800, Chi-Square Test Results for Filters 7-10, N=3

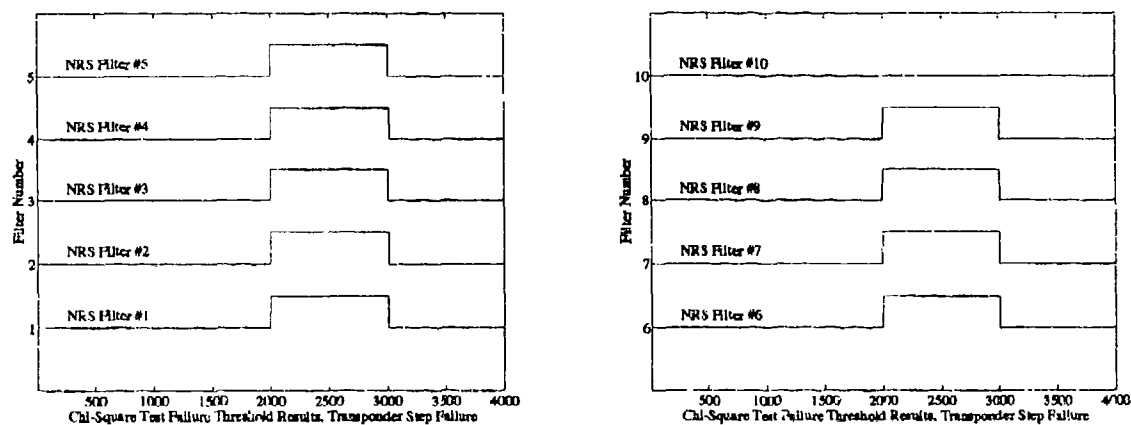


Figure K.6 Threshold Test Results for the Chi-Square Tests on each Filter, Step Bias on Transponder 1

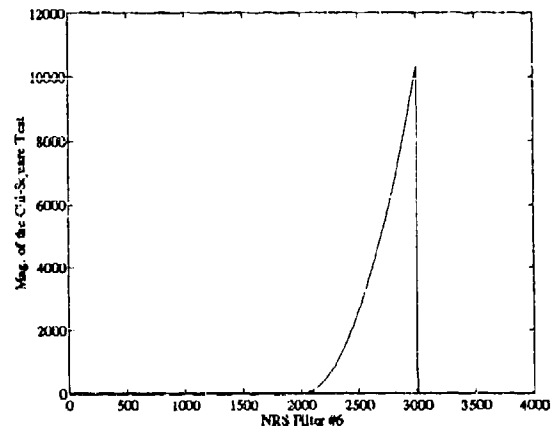
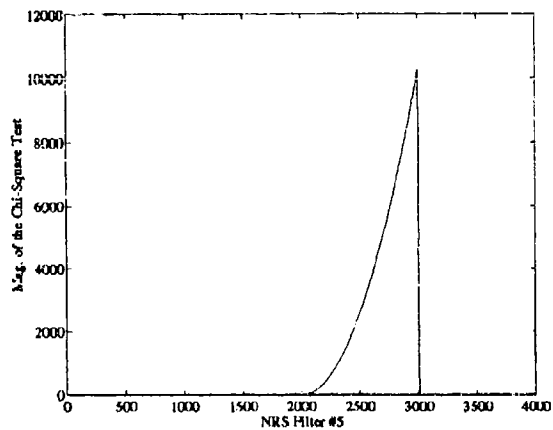
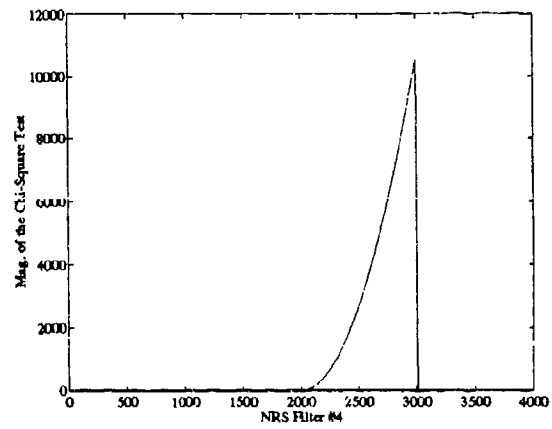
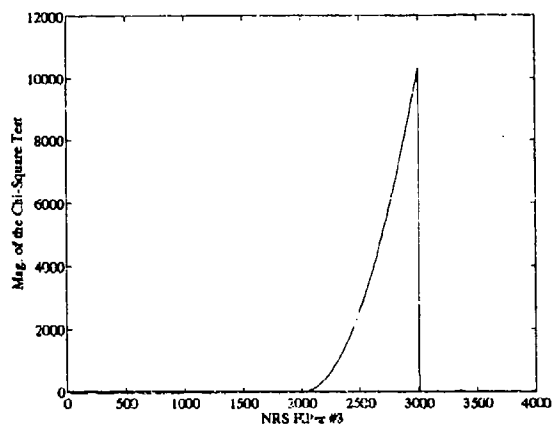
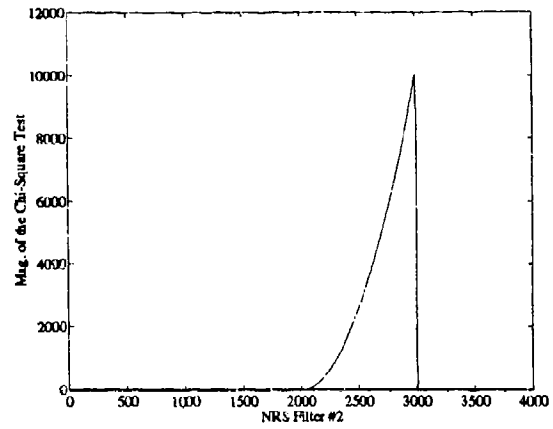
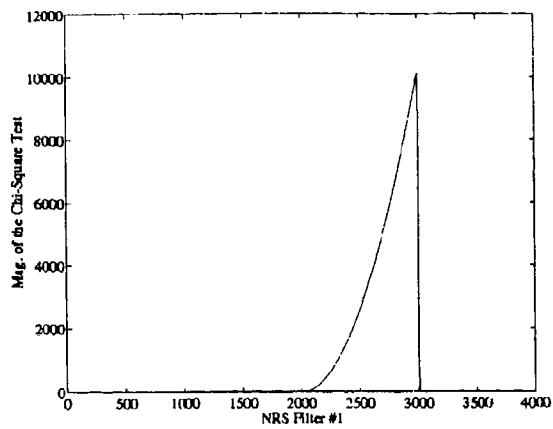


Figure K.7 Transponder 1 *Ramp Offset*=1t, Chi-Square Test Results for Filters 1-6, N=3

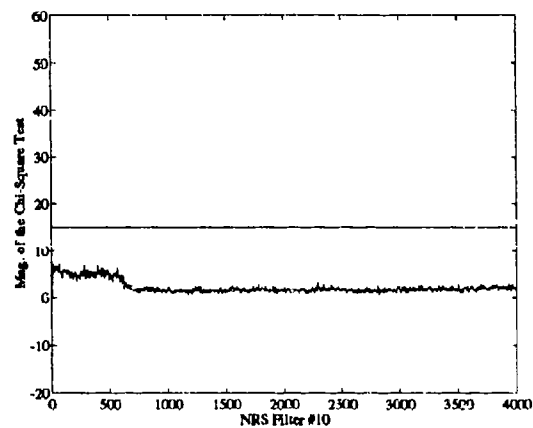
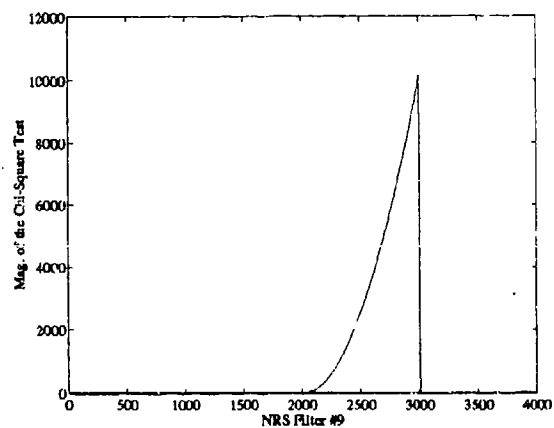
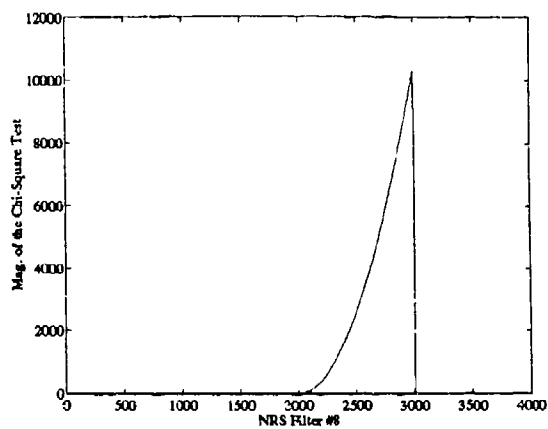
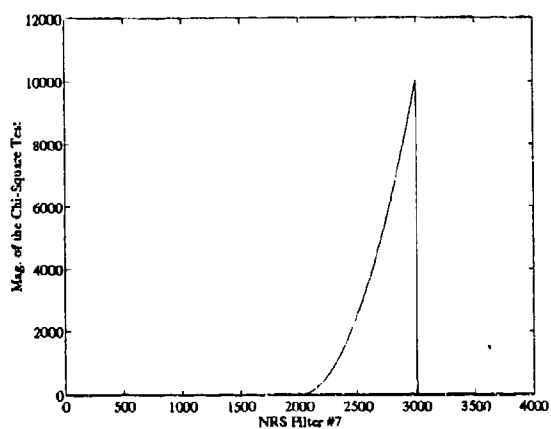


Figure K.8 Transponder 1 *Ramp Offset=1t*, Chi-Square Test Results for Filters 7-10, $N=3$

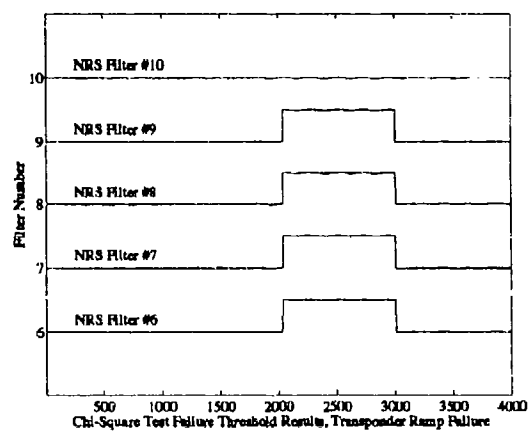
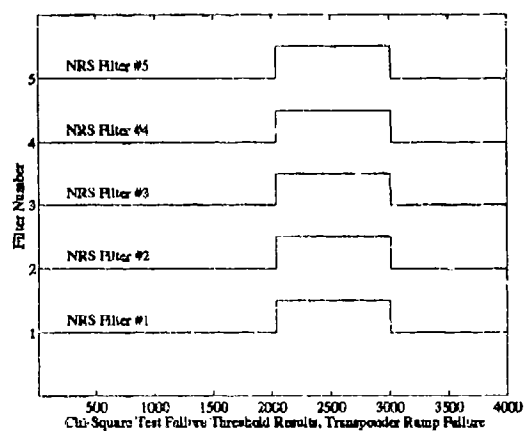


Figure K.9 Threshold Test Results for the Chi-Square Tests on each Filter, Ramp Offset on Transponder 1

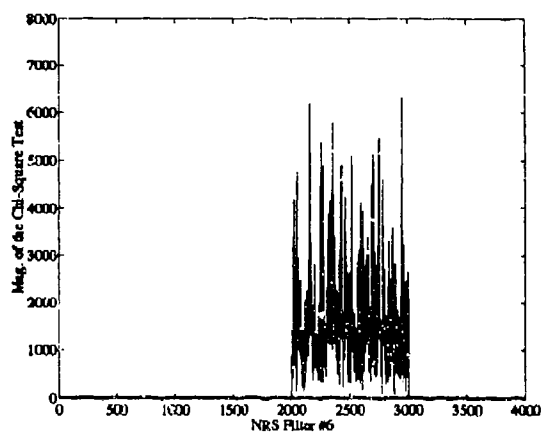
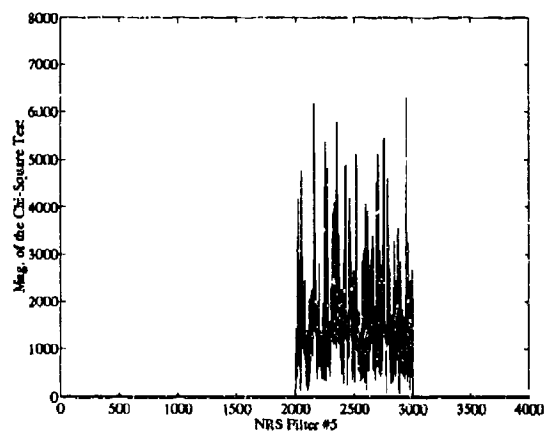
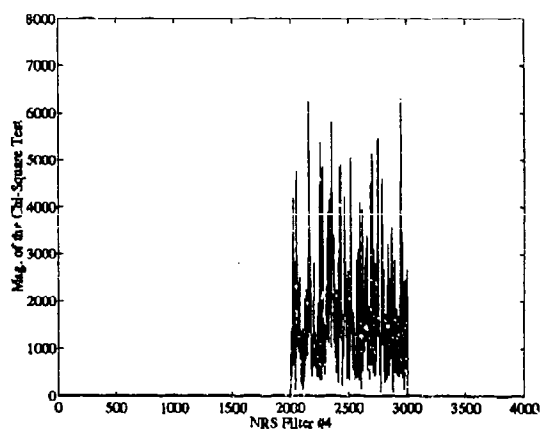
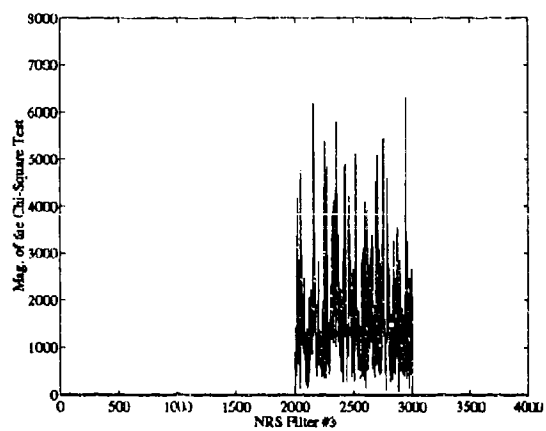
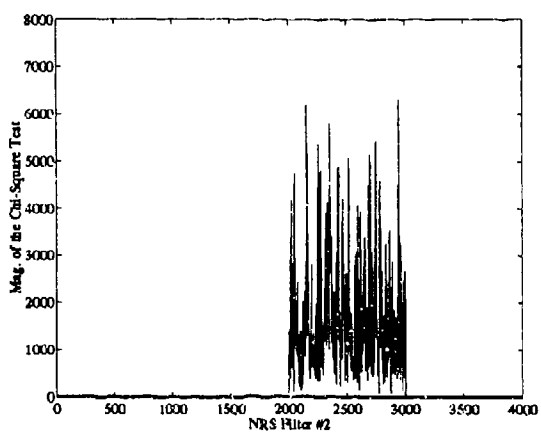
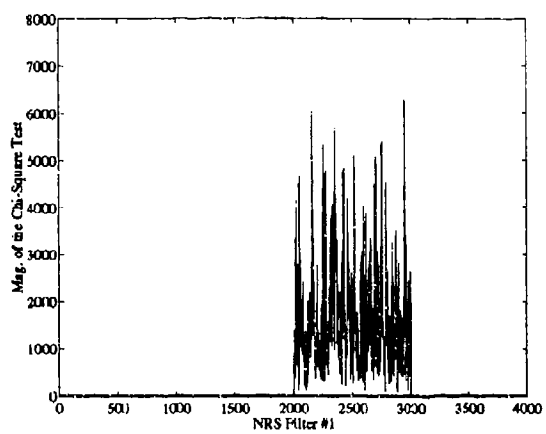


Figure K.10 Transponder 1 *Noise Increase*=150, Chi-Square Test Results for Filters 1-6, N=3

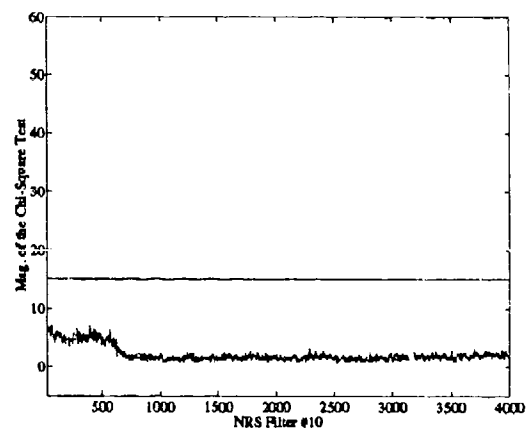
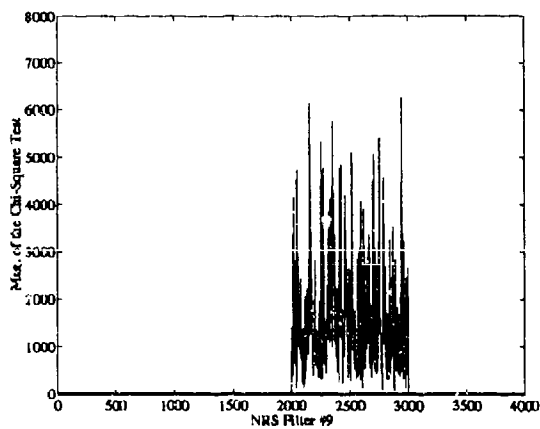
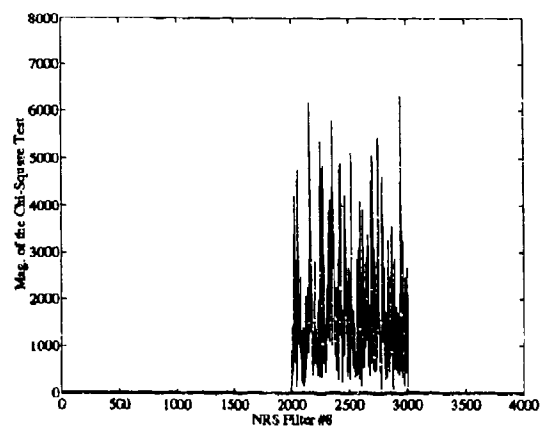
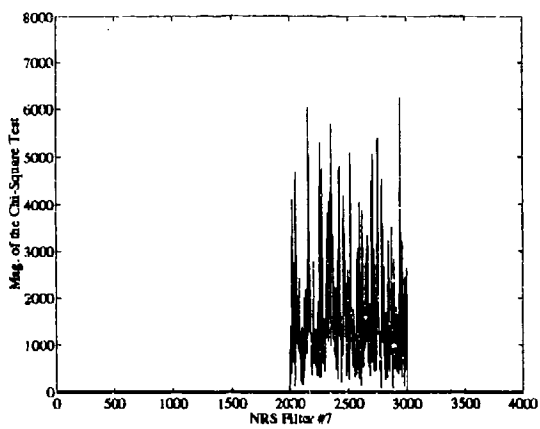


Figure K.11 Transponder 1 *Noise Increase=150*, Chi-Square Test Results for Filters 7-10, $N=3$

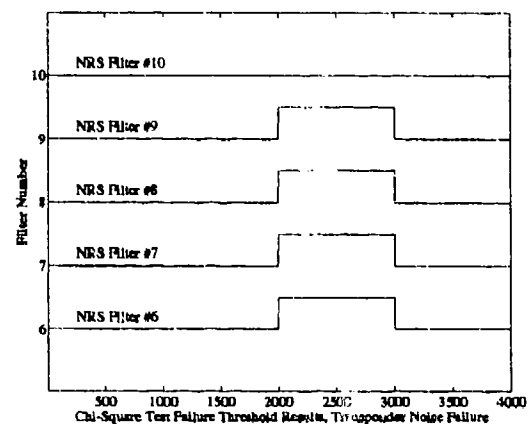
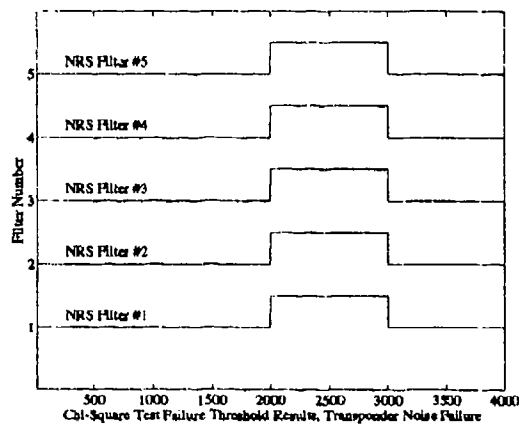


Figure K.12 Threshold Test Results for the Chi-Square Tests on each Filter, Noise Bias on Transponder 1

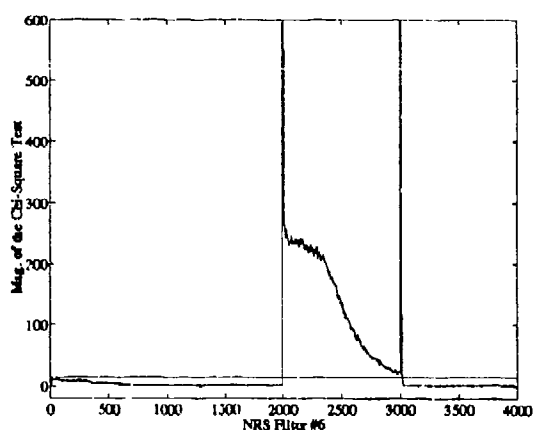
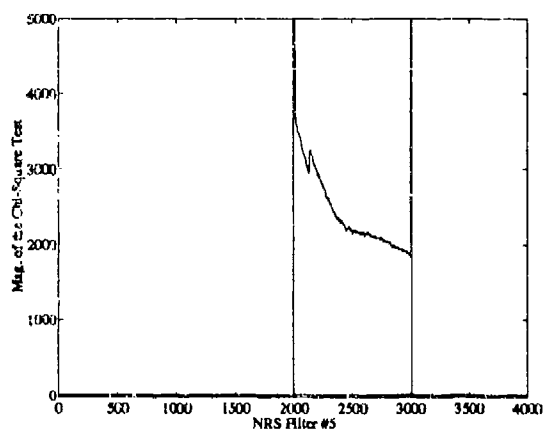
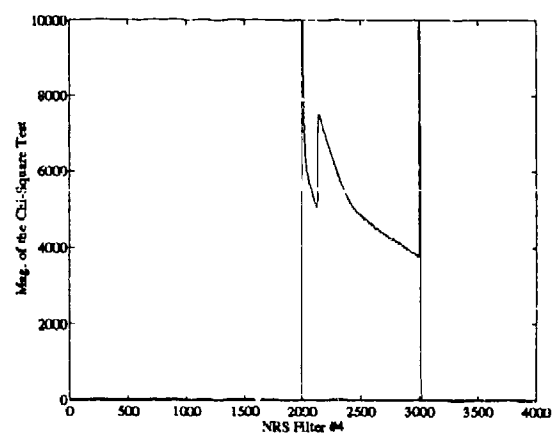
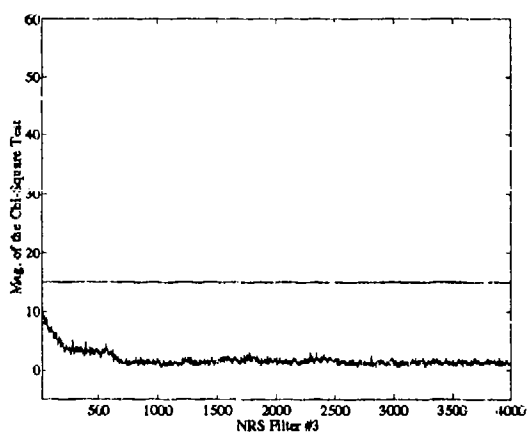
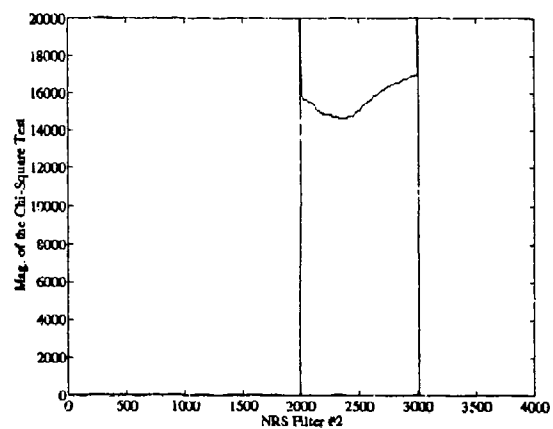
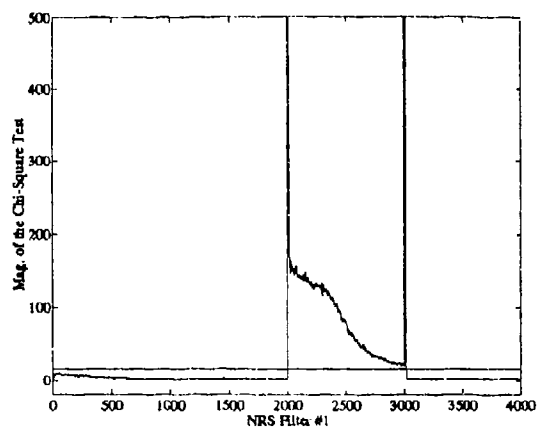


Figure K.13 Satellite 3 Step Bias=3500, Chi-Square Test Results for Filters 1-6, N=3

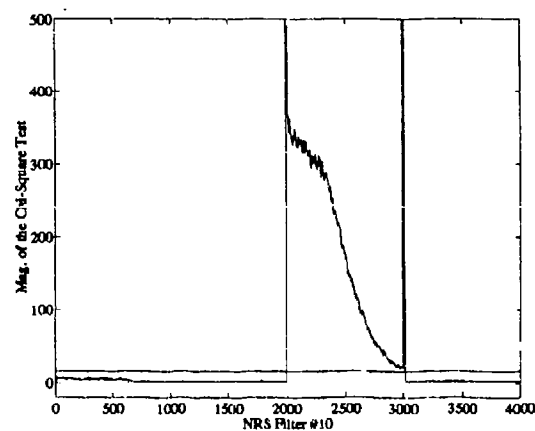
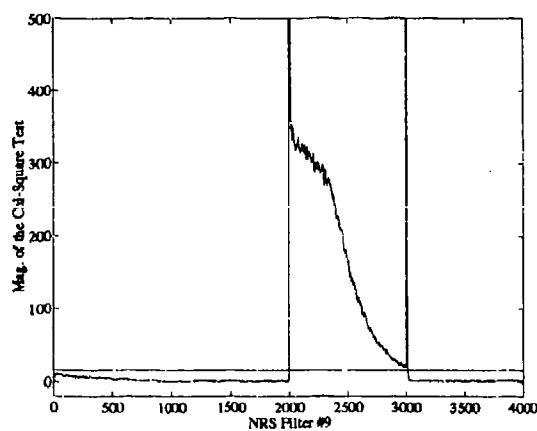
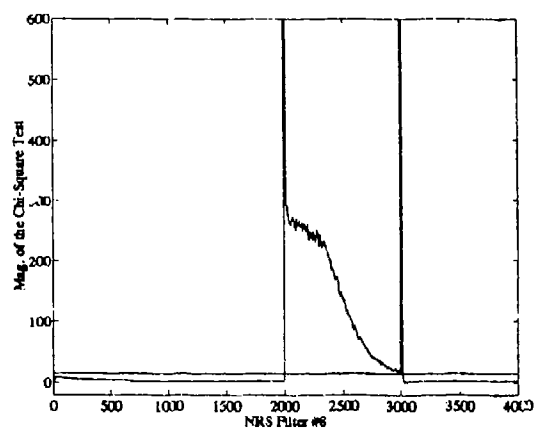
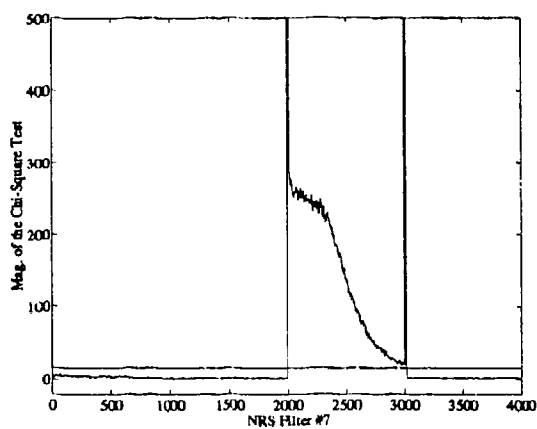


Figure K.14 Satellite 3 Step Bias=3500, Chi-Square Test Results for Filters 7-10, N=3

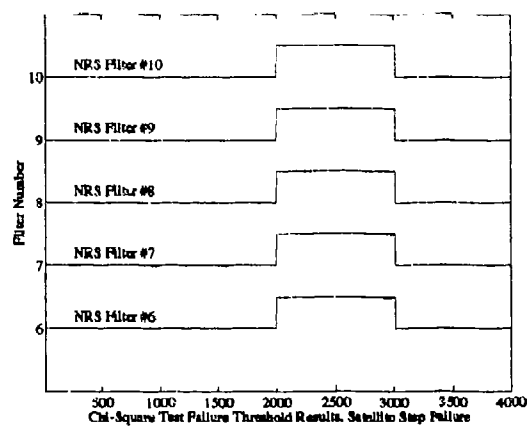
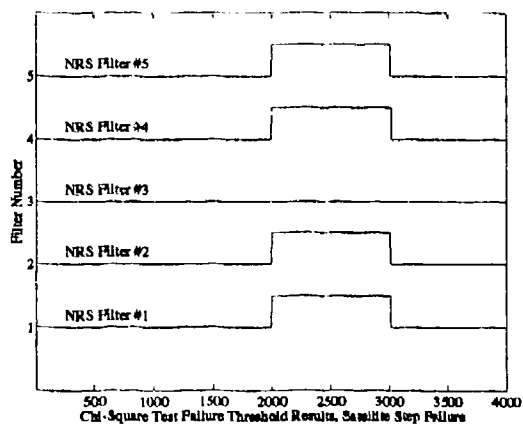


Figure K.15 Threshold Test Results for the Chi-Square Tests on each Filter, Step Bias Run on Satellite 3

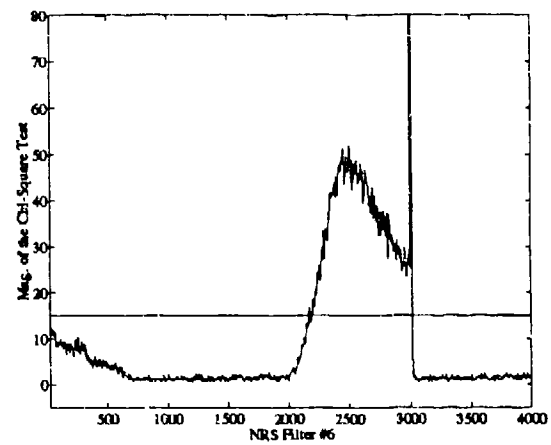
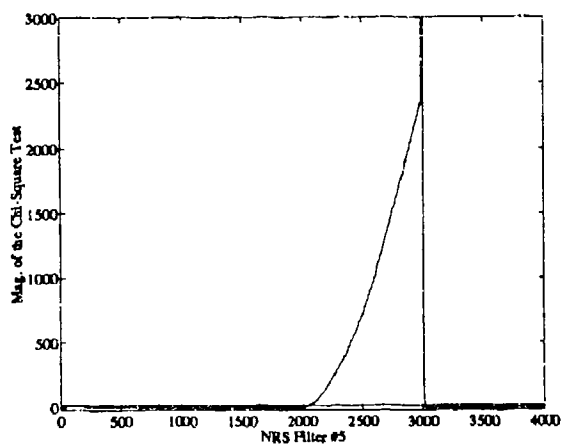
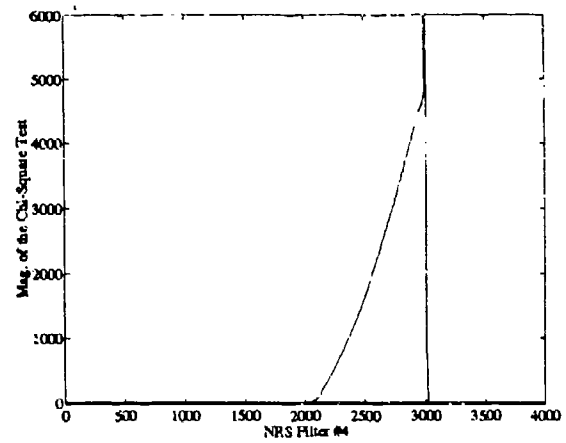
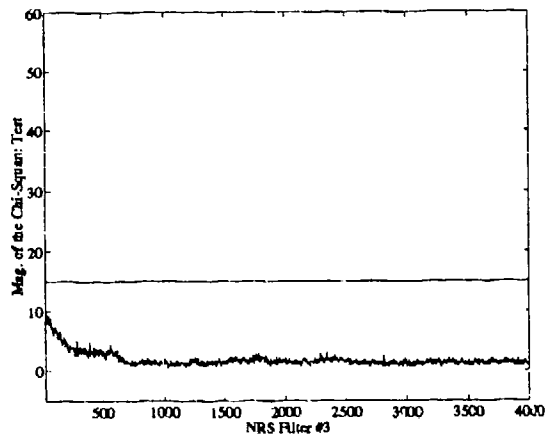
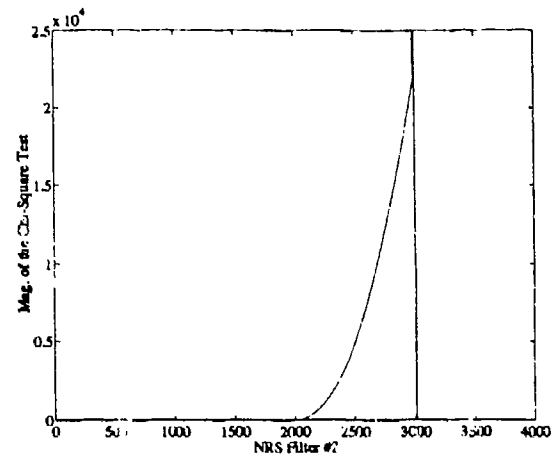
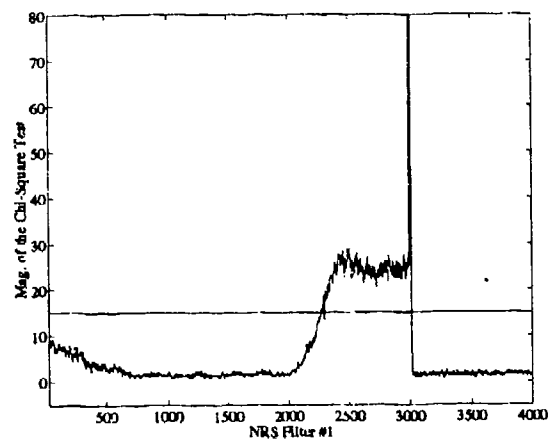


Figure K.16 Satellite 3 *Ramp Offset=4t*, Chi-Square Test Results for Filters 1-6. $N=3$

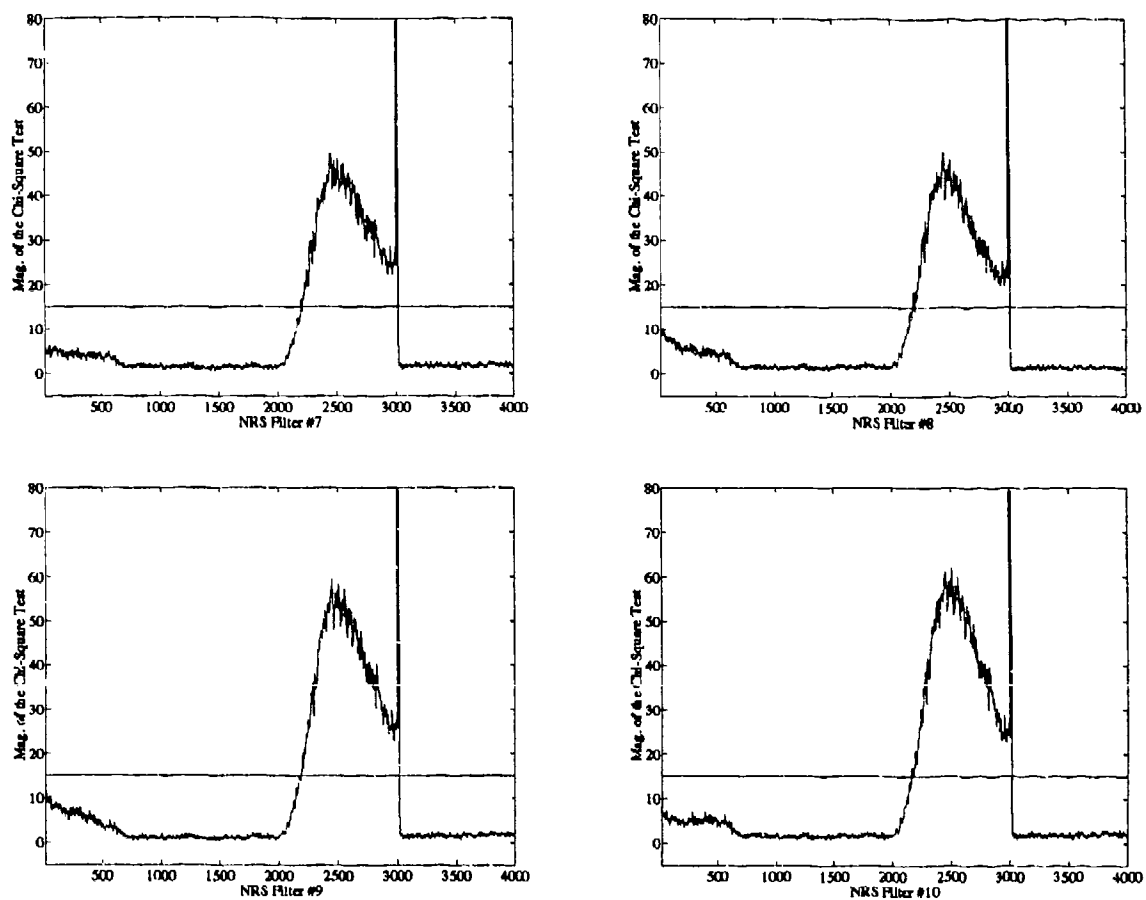


Figure K.17 Satellite 3 Ramp Offset=4t, Chi-Square Test Results for Filters 7-10, N=3

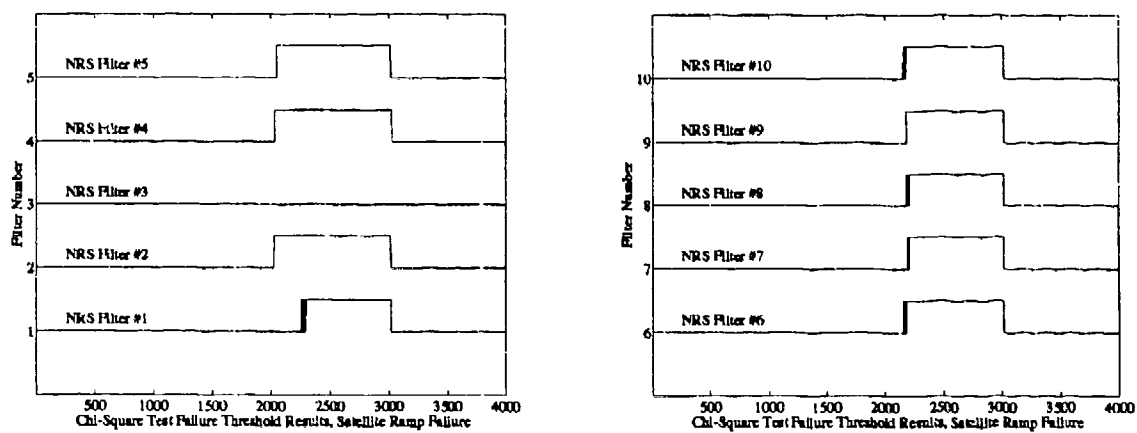


Figure K.18 Threshold Test Results for the Chi-Square Tests on each Filter, Ramp Offset on Satellite 3

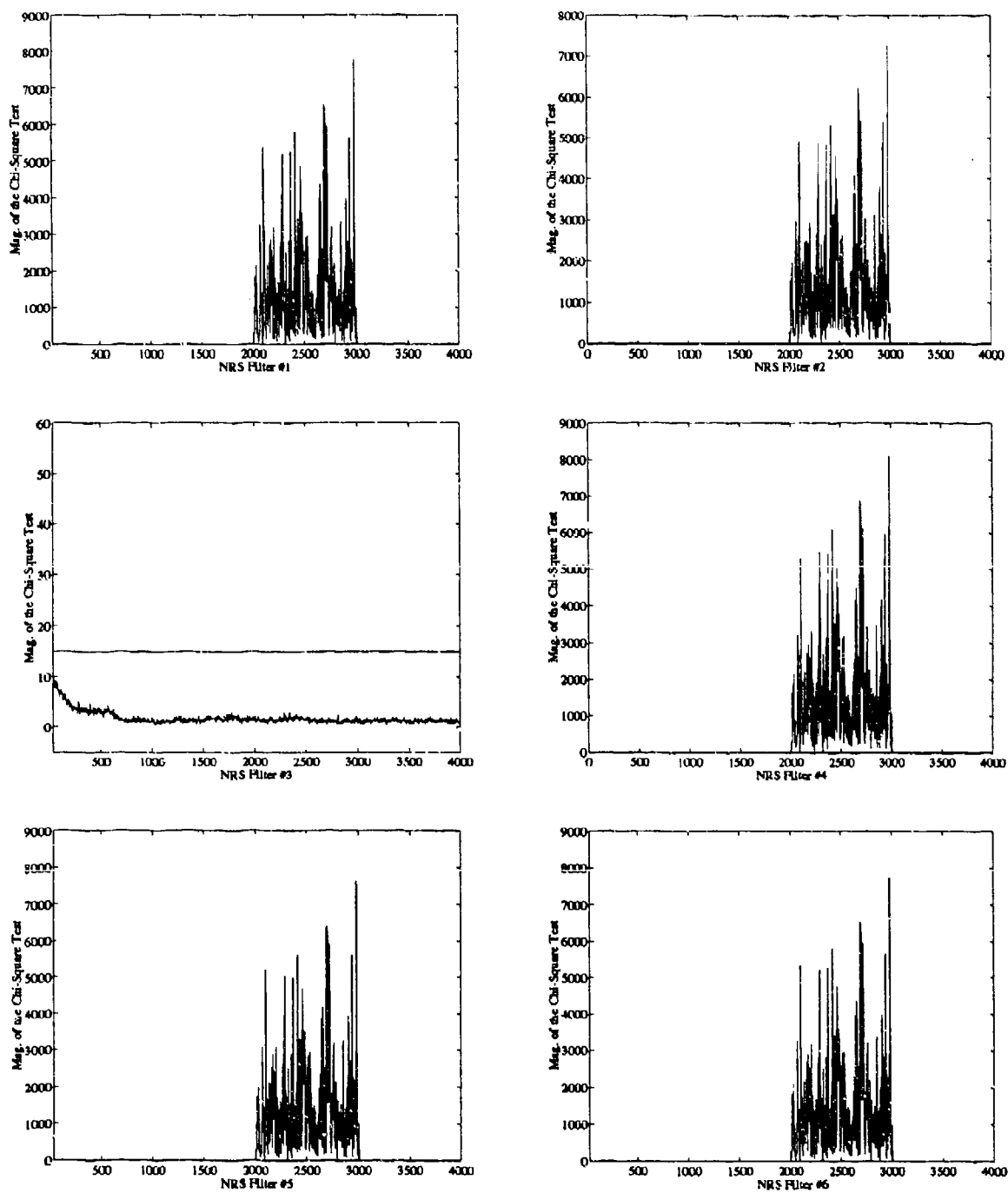


Figure K.19 Satellite 3 *Noise Increase*=300, Chi-Square Test Results for Filters 1-6, N=3

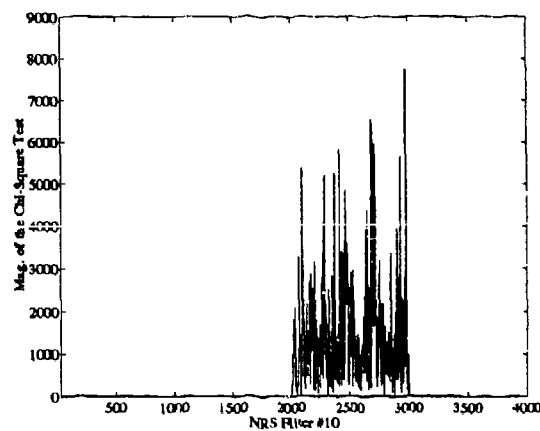
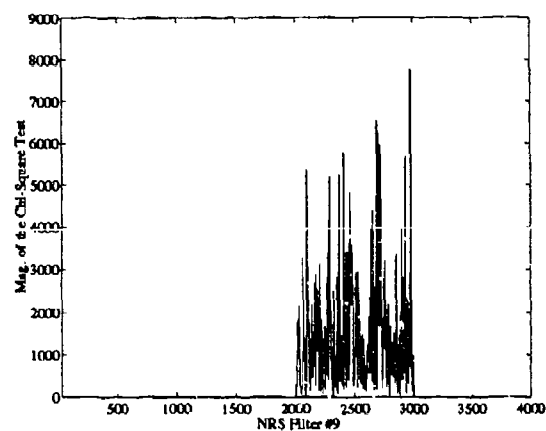
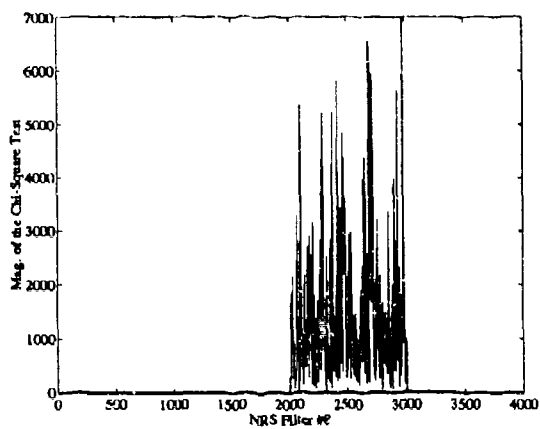
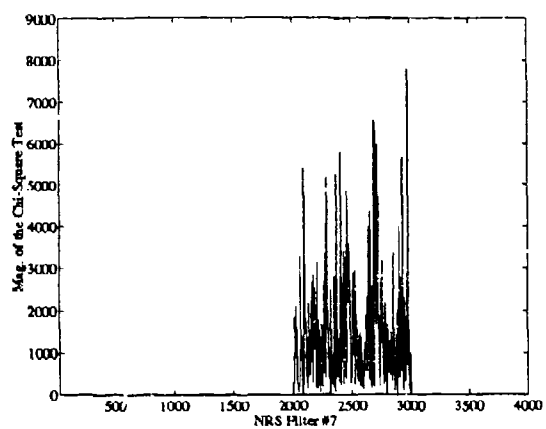


Figure K.20 Satellite 3 *Noise Increase=300*, Chi-Square Test Results for Filters 7-10, $N=3$

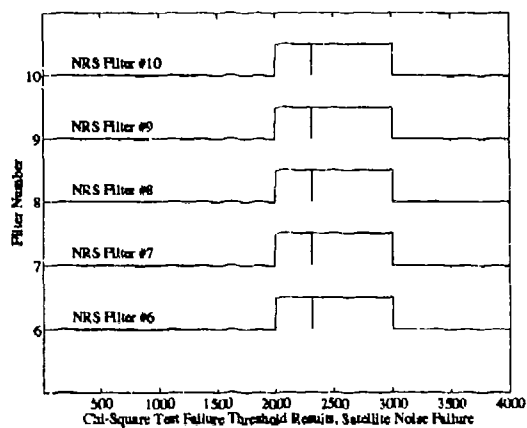
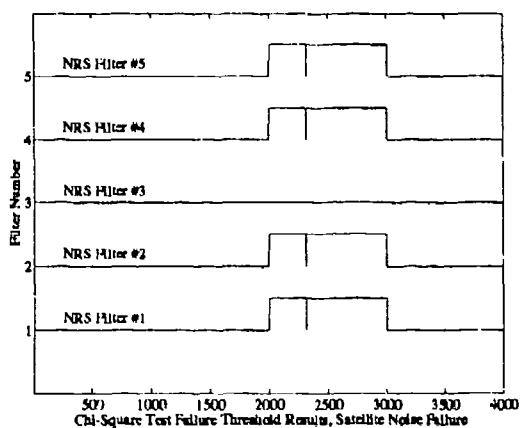
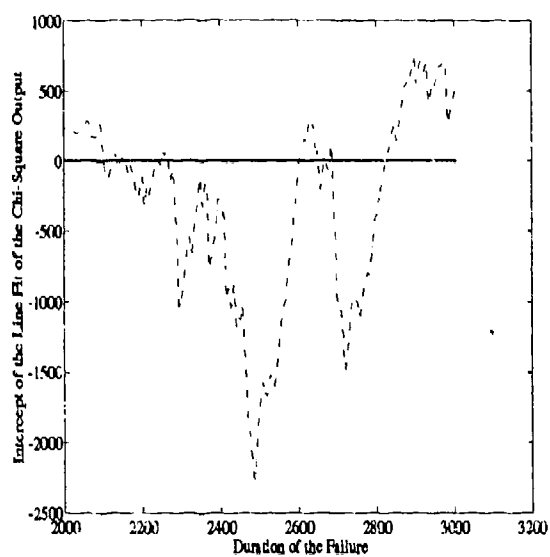


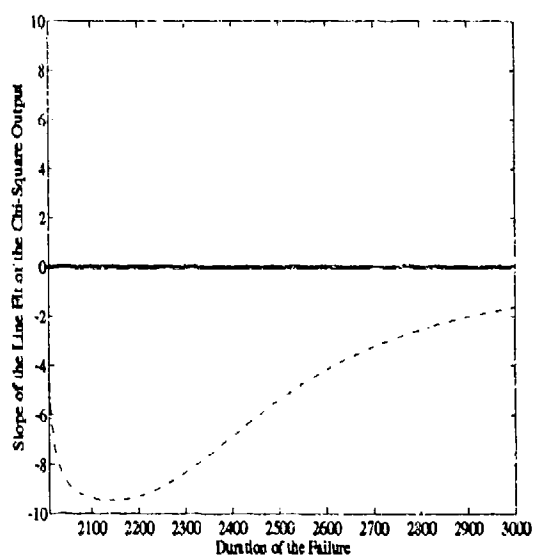
Figure K.21 Threshold Test Results for the Chi-Square Tests on each Filter, Noise Bias on Satellite 3

Appendix L. Slope and Intercept Results of the CSPR Matching Filter

This appendix contains results of a CSPR matching filter simulation for each of the six failures used in this thesis. The magnitude and duration of the simulated failures are identical to those used for the MNRS simulation runs. Two variables are documented: the slope and intercept of a line fit of the Chi-Square test. As stated in Chapter IV, the slope and intercept are plotted as a function of time over the duration of the detected failure. This duration is defined as all points in time where the Chi-Square test exceeds the failure threshold. The first order line fit of the Chi-Square data is performed in MATLAB [8]. The intercept and slope for the three transponder failures (step, ramp, and noise) are presented first, followed by plots for the three satellite failures (same order). Analysis for the CSFR simulation is located in Section 4.5.

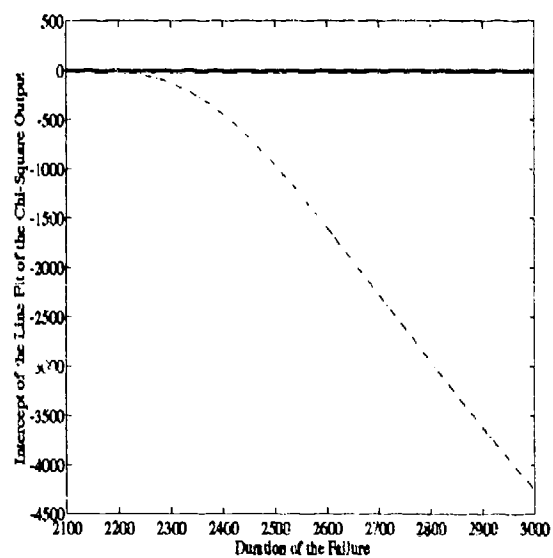


Intercept of the Line Fit

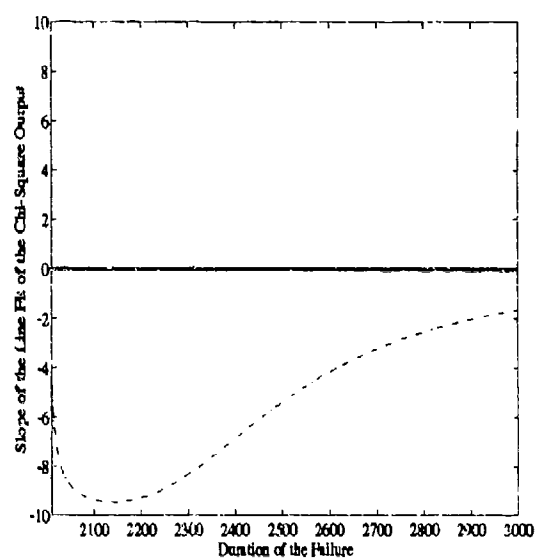


Slope of the Line Fit

Figure L.1 NRS Filter 1 Chi-Square Line Fit Data for a Transponder Step Bias

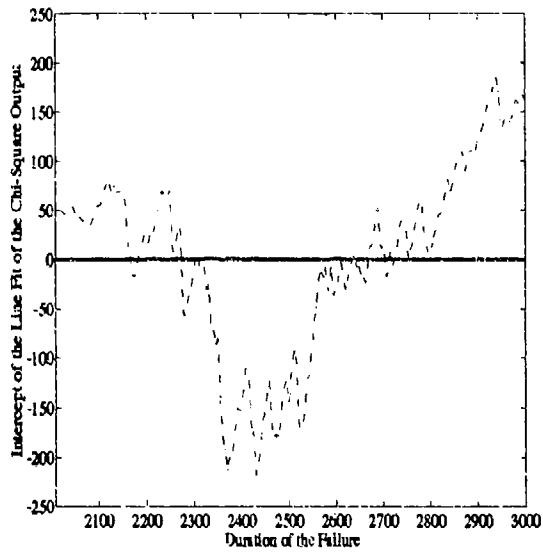


Intercept of the Line Fit

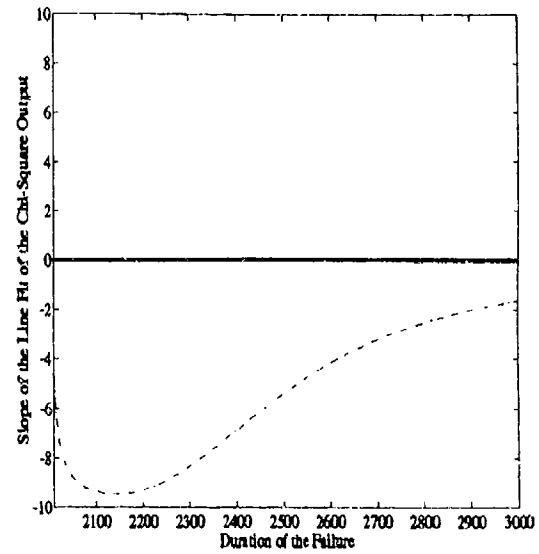


Slope of the Line Fit

Figure L.2 NRS Filter 1 Chi-Square Line Fit Data for a Transponder Offset Bias

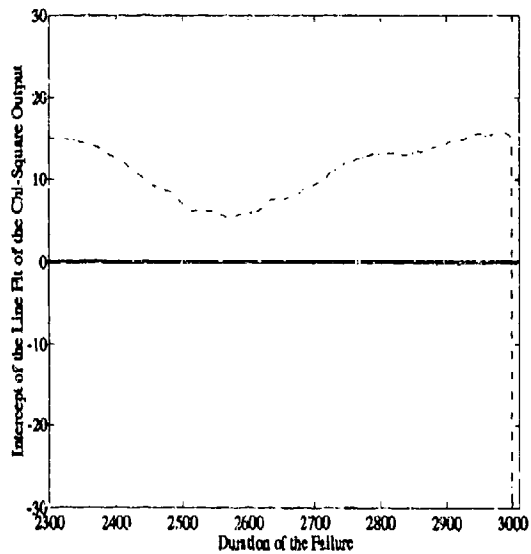


Intercept of the Line Fit

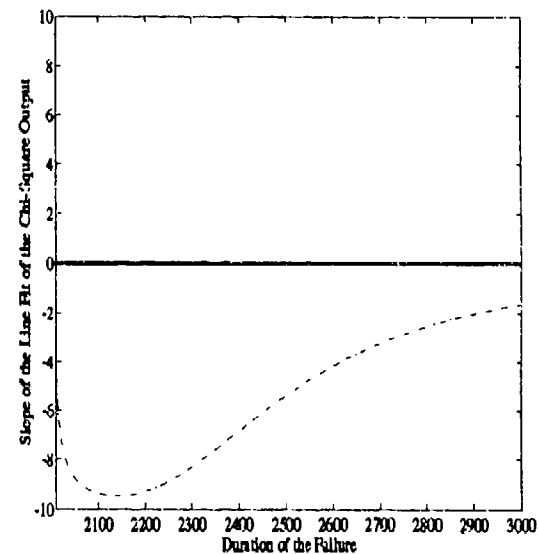


Slope of the Line Fit

Figure L.3 NRS Filter 1 Chi-Square Line Fit Data for a Transponder Noise Increase

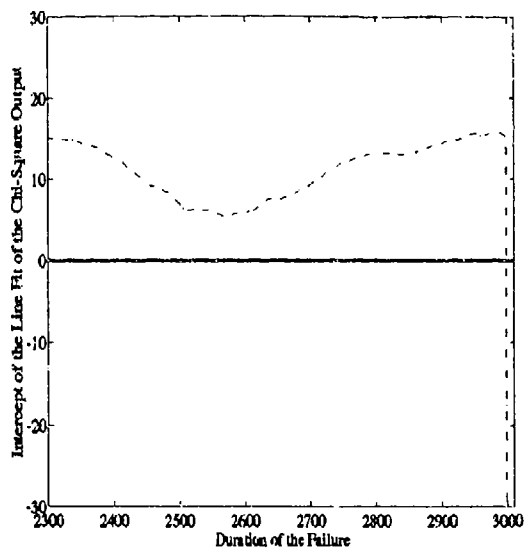


Intercept of the Line Fit

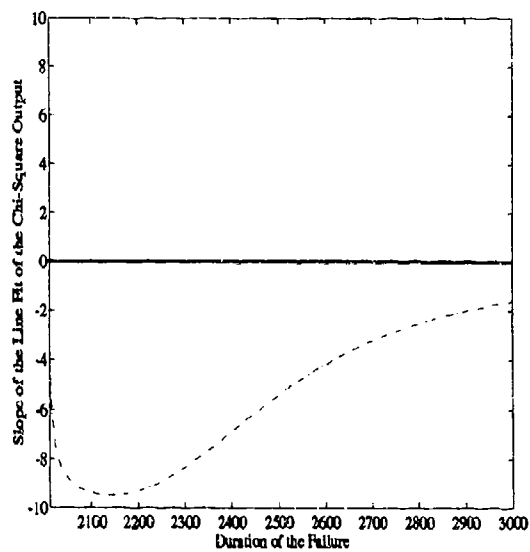


Slope of the Line Fit

Figure L.4 NRS Filter 1 Chi-Square Line Fit Data for a Satellite Step Bias

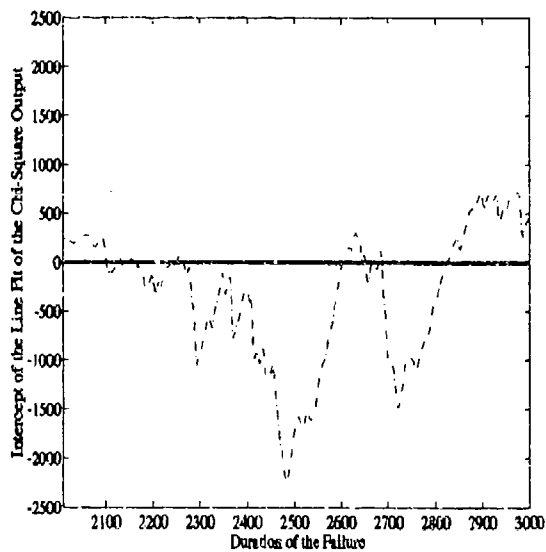


Intercept of the Line Fit

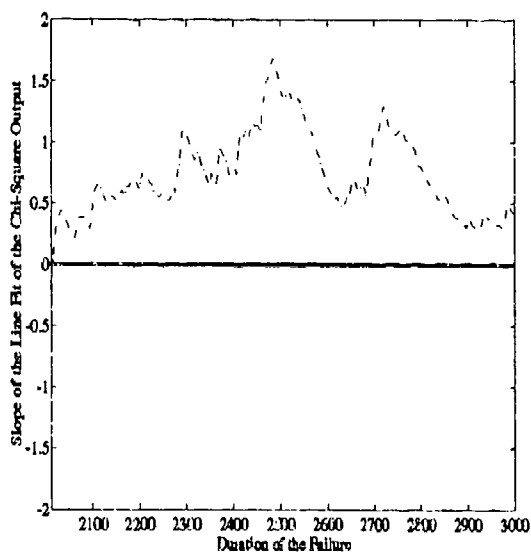


Slope of the Line Fit

Figure L.5 NRS Filter 1 Chi-Square Line Fit Data for a Satellite Ramp Offset



Intercept of the Line Fit



Slope of the Line Fit

Figure L.6 NRS Filter 1 Chi-Square Line Fit Data for a Satellite Noise Increase

Bibliography

1. Britting, Kenneth R. *Inertial Navigation Systems Analysis*. Wiley-Interscience, 1971.
2. Grondin, P., et al. *Multiple Model Failure Detection for an Integrated Navigation System*. Technical Report, AFIT, WPAFB, Ohio, Spring 1993.
3. Gustafson, J., et al., "Corrections to the Litton documentation on the LN-93 Model," May 1993. Unpublished discussion on the Litton Barometric Altimeter Error Model.
4. Knudsen, L. *Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation System Inertial Navigation Unit*. Technical Report, Litton Guidance and Control Systems, January 1985. DID No. DI-S-21433 B/T: CDRL No. 1002.
5. Kobylarz, T., et al. *Satellite Orbit Project*. Technical Report, AFIT, WPAFB, Ohio, Spring 1993.
6. Lewantowicz, Zdzislaw H. and Randy Paschall. *Deep Integration of GPS, INS, SAR, and Other Sensor Information*. Technical Report, Systems Avionics Division, Avionics Directorate, Wright Laboratory, 1993. Unpublished.
7. Martin, E. H. "GPS User Equipment Error Models," *The Institute of Navigation, Volume I*:109-118 (1980).
8. The MathWorks, Inc, 21 Eliot Street, Sout Natick, MA 01760. *Matlab*, December 1992. Version 4.0a.
9. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, I*. Academic Press, Inc., 1979.
10. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, III*. Academic Press, Inc., 1982.
11. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, II*. Academic Press, Inc., 1982.
12. Musick, Stanton H. *PROFGEN - A Computer Program for Generating Flight Profiles*. Technical Report, Air Force Avionics Laboratory, WPAFB, Ohio, November 1976. AFAL-TR-76-247, DTIC ADA034993.
13. Musick, Stanton H. *MSOFE - Multimode Simulation for Optimal Filter Evaluation*. Technical Report, Air Force Avionics Laboratory, WPAFB, Ohio, October 1980. AFWAL-TR-88-1138.
14. Negast, William Joseph. *Incorporation of Differential Global Positioning System Measurements Using and Extended Kalman Filter for Improved Reference System Performance*. MS thesis, Air Force Institute of Technology, WPAFB, OH, December 1991.
15. Riggins, Robert N, Jr. *Detection and Isolation of Plant Failures in Dynamic Systems*. PhD dissertation, University of Michigan, Ann Arbor, MI, 1991.
16. Schwartz, Howard M. "Sensititivity Analysis of an Integrated NAVSTAR GPS/INS Navigation System to Component Failure," *Journal of The Institute of Navigation, Vol. 30* (1983-1984).

17. Snodgrass, Faron Britt. *Continued Development and Analysis of a New Extended Kalman Filter for the Completely Integrated Reference Instrumentation System (CIRIS)*. MS thesis, Air Force Institute of Technology, WPAFB, OH, March 1990.
18. Stacey, Richard D. *A Navigation Reference System Using Global Positioning System and Transponder Aiding*. MS thesis, Air Force Institute of Technology, WPAFB, OH, March 1991.
19. Vasquez, Juan R. *Detection of Spoofing, Jamming, or Failure of a Global Positioning System (GPS)*. MS thesis, Air Force Institute of Technology, WPAFB, OH, December 1992.
20. Willsky, A. and H. Jones. "A Generalized Likelihood Ratio Approach to State Estimation in Linear Systems Subject to Abrupt Changes," *The Analytic Sciences Corporation, Reading, Mass., under USAF Contract No. F04701-74-C-0095* (1976).
21. Willsky, Alan S. "A survey of design methods for failure detection in dynamic systems," *Automatica, Vol. 12*, 601-611 (1976).

Vita

Second Lieutenant William Boardman Mosle III was born 30 January 1970 in Pittsburgh, Pennsylvania. He spent his entire youth in Pittsburgh, graduating from Shady Side Academy in June, 1988. He continued his academic career at Duke University with the aid of an Air Force Reserve Officer Training Corps (ROTC) scholarship. The collegiate experience was highlighted by the "Back-to-Back" 1991-1992 Duke Men's NCAA Division I Basketball Championships. Duke also provided the opportunity for William to meet his fiancé, fellow Blue Devil, Jennifer Dennis. Upon graduation from the Duke undergraduate engineering program in May 1992, he received his commission as an Air Force officer. Promptly after graduation, William returned to his career as a professional student at the Air Force Institute of Technology (AFIT). From May 1992-December 1993, he pursued his Masters of Science in Electrical Engineering (MSEE) degree in the study of navigation and control systems. In December 1993, his perpetual academic career came to an end with his assignment to the 46th Test Wing (AFMC), Holloman AFB, N.M., where he is currently working.

Permanent address: 1100 Liberty Avenue
Pittsburgh, PA 15222

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1993		3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE FAILURE DETECTION, ISOLATION, AND RECOVERY IN AN INTEGRATED NAVIGATION SYSTEM				5. FUNDING NUMBERS	
6. AUTHOR(S) William B. Mosle III, Second Lieutenant, USAF					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/93D-28	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) 46th Test Wing Holloman AFB, NM, 88330-5000				10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; Distribution Unlimited				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An Inertial Navigation System(INS), the Global Positioning System(GPS), and a ground based transponder system(RRS) can all be used to provide the user with a navigation solution. Yet by integrating these three navigation systems with an extended Kalman filter(EKF), a navigation solution is attained that benefits from the information of all three subsystems. This research develops a multiple model EKF failure detection, isolation, and recovery(FDIR) algorithm using a Chi-Square failure test to provide robust navigation solution to measurement failures. The algorithm specifically counters failures in the GPS and RRS range measurements. Analysis is conducted using a Kalman filter development package known as the Multimode Simulation for Optimal Filter Evaluation (MSOFE). Both a large order <i>truth</i> model for the navigation system (in which a full 24 satellite constellations is modeled) and a reduced-order Kalman filter are developed. Results suggest that the multiple model algorithm can correct for all single measurement failures.					
14. SUBJECT TERMS Integrated Navigation System, Extended Kalman Filter, Failure Detection, Robust Navigation				15. NUMBER OF PAGES 238	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL		

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.